

1^η Διδακτική ώρα

Γνωριμία με το περιβάλλον

- Παρουσίαση του περιβάλλοντος από τον καθηγητή.
- Παρουσίαση χρήσης από τον καθηγητή.
- Δημιουργία λογαριασμών μαθητών.
- Εξοικείωση με το περιβάλλον του Pencil code.

2η Διδακτική ώρα

Εντολές εξόδου

ΣΤΟΧΟΣ:

Στόχοι του μαθήματος είναι οι μαθητές να:

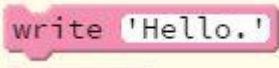
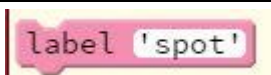
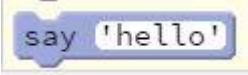

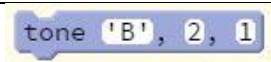
- αναφέρουν τις εντολές εξόδου που διαθέτει το περιβάλλον του pencil code
- περιγράφουν τη χρήση των εντολών εξόδου
- χρησιμοποιούν τις εντολές εξόδου

Εντολές εξόδου και αριθμητικές πράξεις

Μια βασική δυνατότητα που έχουν οι υπολογιστές και κατ' επέκταση τα προγραμματιστικά περιβάλλοντα είναι η εκτέλεση αριθμητικών πράξεων. Ιστορικά μάλιστα οι υπολογιστές χρησιμοποιήθηκαν κυρίως για την εκτέλεση σύνθετων υπολογισμών. Φυσικά σε ένα προγραμματιστικό περιβάλλον θα πρέπει να υπάρχει δυνατότητα να πάρουμε με κάποιο τρόπο το αποτέλεσμα μια πράξης. Το έργο αυτό το αναλαμβάνουν οι εντολές εξόδου. Οι εντολές εξόδου αναλαμβάνουν να εξάγουν κάτι είτε στην οθόνη είτε στα ηχεία είτε σε χαρτί. Κάθε εφαρμογή διαθέτει εντολές για εισαγωγή δεδομένων από το πληκτρολόγιο και το ποντίκι και εξαγωγή του αποτελέσματος σε οθόνη και ηχεία.

Οι εντολές εξόδου στην οθόνη βρίσκονται στην οικογένεια πλακιδίων Text και είναι χρωματισμένες ροζ, ενώ οι εντολές εξόδου στα ηχεία βρίσκονται στην οικογένεια πλακιδίων Sound και είναι γαλάζιο-μωβ.

Οι κυριότερες εντολές είναι :

write	Εμφανίζεται το κείμενο μας στην οθόνη	
label	εμφανίζεται το κείμενο μας στη θέση που είναι η χελώνα. Συντάσσεται όμοια με τη write	
say	Έξοδο στα ηχεία. Συντάσσεται όμοια με τη write.	
play	παίζει μελωδία	
tone	Ακούγεται τόνος	

Υπάρχουν και ορισμένες εντολές εξόδου οι οποίες είναι διαθέσιμες στο περιβάλλον εντολών (CoffeeScript) και δεν υπάρχουν στα αρχικές οικογένειες πλακιδίων. Αν όμως την χρησιμοποιήσετε σε περιβάλλον εντολών τότε το περιβάλλον δημιουργεί ένα αντίστοιχο πλακίδιο. Το φαινόμενο αυτό ισχύει και για άλλες εντολές της CoffeeScript.

Για παράδειγμα η εντολή `alert 'μήνυμα'` η οποία εμφανίζει το 'μήνυμα' σε ένα ειδικό παράθυρο.

Πλακίδιο `alert` εν υπάρχει στα αρχικά πλακίδια, αν όμως χρησιμοποιήσετε την εντολή στο περιβάλλον του κώδικα και στη συνέχεια μεταβείτε στο περιβάλλον πλακιδίων θα διαπιστώσετε ότι έχει εμφανιστεί ένα πλακίδιο με όνομα `alert`.

Χρήση της εντολής `write`

I. Εμφάνιση αποτελέσματος αριθμητικών πράξεων (αριθμητική έκφραση)

Οι αριθμητικές πράξεις πραγματοποιούνται μεταξύ αριθμών με τη βοήθεια των γνωστών συμβόλων `+` για πρόσθεση, `-` για αφαίρεση, `*` για πολλαπλασιασμό, `/` για διαίρεση

Για την εμφάνιση του αποτελέσματος χρησιμοποιούμε την εντολή `rite` και δίπλα την αριθμητική έκφραση Πχ `write 20 / 4` (αποτέλεσμα 5)

Ισχύει η προτεραιότητα πράξεων όπως την ξέρουμε στα μαθηματικά αλλά σε περιπτώσεις σύνθετων πράξεων αν υπάρχει αμφιβολία καλό είναι να χρησιμοποιούνται παρενθέσεις.

Εκτός από αριθμητικές εκφράσεις δίπλα στην `write` μπορούν να χρησιμοποιηθούν μαθηματικές συναρτήσεις Πχ `write round 6,5`

II. Εμφάνιση Αλφαριθμητικών

Τα *αλφαριθμητικά* (strings), μπορεί να είναι φράσεις, λέξεις ή και αριθμοί που δε γίνονται αριθμητικές πράξεις.

-Αν θέλουμε η εμφάνιση να γίνει σε μια παράγραφο τοποθετούμε το αλφαριθμητικό μέσα σε μονά ή διπλά εισαγωγικά.

Π.χ `write 'Είμαι μαθητής της Γ γυμνασίου. Είμαι 14 χρονών. Μου αρέσει το μπάσκετ. Μένω στην Ελλάδα. Αγαπημένο μου χρώμα είναι το κόκκινο'`

-Αν θέλουμε να έχουμε περισσότερες παραγράφους (στην οθόνη) θα πρέπει εκεί που αρχίζει και τελειώνει κάθε παράγραφος να τοποθετούμε τις ενδείξεις `<r>` και `</r>`.

'Υπάρχουν διαθέσιμες και άλλες ενδείξεις (τύπου HTML) για την μορφοποίησης κειμένου όπως τα `<i>..</i>` για πλάγια γραφή ή ` ...` για έντονα γράμματα (bold).

III. Ενώνοντας αλφαριθμητικά

Αν θέλουμε να συνενώσουμε δύο ή περισσότερα α αλφαριθμητικά βάζουμε το σύμβολο `+` ανάμεσα σε αυτά. Πχ `write 'a' + '3'` θα δώσει `a3`

Παραδείγματα

<u>Εντολή <code>write</code></u>	<u>αποτέλεσμα</u>
<code>write 'καλημέρα'</code>	καλημέρα

<pre>write '<p> Σήμερα είναι Δευτέρα. </p> <p> Είναι Νοέμβριος.</p> <p> Πηγαίνω στη Γ γυμνασίου.</p>'</pre>	<p>Σήμερα είναι Δευτέρα.</p> <p>Είναι Νοέμβριος.</p> <p>Πηγαίνω στη Γ γυμνασίου.</p>
<pre>write '10+5'</pre>	10 + 5
<pre>write 10+5</pre>	15
<pre>write 2+3*5</pre>	17
<pre>write (2+3)*5</pre>	25
<pre>write 'a' + 3 + 5</pre>	a35
<pre>write round 6,7</pre>	6
<pre>write abs -7</pre>	7
<pre>write max 10, 3</pre>	10
<pre>write "12 + 8 = #{12 + 8}"</pre>	12 + 8 = 20
<pre>write '5 + 3 δεν είναι' + Number('5') + Number('3')</pre>	5 + 3 δεν είναι 53
<pre>write 3 > 2</pre>	true
<pre>write '<i> Μου αρέσουν τα <u> φρούτα </u> </i>'</pre>	Μου αρέσουν τα <u>φρούτα</u>
<pre>write "Σ'ευχαριστώ πολύ. "</pre>	Σ' ευχαριστώ πολύ.
<pre>write 'Ο Λεωνίδας είπε: " Μολών λαβέ!'"</pre>	Ο Λεωνίδας είπε: " Μολών λαβέ!"

Σχέδιο μαθήματος:

Μοιράζεται το φύλλο εργασίας και ζητείται από τους μαθητές να το συμπληρώσουν

Οι μαθητές δουλεύουν είτε μόνος του ο καθένας είτε συνεργατικά ανάλογα τον αριθμό των μαθητών και των διαθέσιμων υπολογιστών.

Σε κάθε περίπτωση επιτρέπεται η επικοινωνία ανάμεσα στους μαθητές για την πραγματοποίηση του φύλλου εργασίας.

Ο διδάσκων κατά τη διάρκεια του μαθήματος κάνει ερωτήσεις στους μαθητές, στο να διαπιστωθεί αν αντιληφθήκαν τη διαφοροποίηση στη σύνταξη των εντολών.

Σχολείο Τμήμα..... Αρ. καταλόγου.....

Όνοματεπώνυμο..... Ημερομηνία.....

Φύλλο εργασίας -Εντολές εξόδου: Κείμενο, αριθμοί, ήχος

Σημείωση: Η εντολή write και label υπάρχουν στην ομάδα πλακιδίων Text

Η εντολή say, play και tone υπάρχουν στην ομάδα πλακιδίων Sound

☞ Ποιο είναι το αποτέλεσμα (Output) των επόμενων εντολών;

Εντολή	αποτέλεσμα	πιο εύκολη είναι η χρήση πλακιδίων (Π) ή κώδικα (Κ);
write 'καλημέρα'		
write 'προσθέτω 3 + 5'		
write 3+5		
write 2+4*5		
write (2 + 4)/3		
write '3' + '5'		
write Number('3') + Number('5')		
write "3 + 5 ισούται με #{3 + 5}"		
write 'Είμαι μαθητής Γυμνασίου'		
write ' <p>Σήμερα έχει κρύο. </p> <p> Είναι Χειμώνας.</p> '		
alert 'καλημέρα'		
label 'καλησπέρα'		
say 'My name is John'		
play 'EDCDEEEzDDDzEEE'		
tone 'A'		

☞ Δώστε τις κατάλληλες εντολές ώστε να εμφανιστούν, με τη μορφοποίηση που σας ζητείται, τα εξής:

	Εντολή	Ποιό περιβάλλον χρησιμοποίησε; Με Πλακίδια (Π) ή με Κώδικα (Κ)
Η λέξη ΕΛΛΑΔΑ (με έντονη γραφή)		
Το γινόμενο του αριθμού 27 με τον αριθμό 35		
Να γραφεί ως ταμπέλα πάνω στη χελώνα η φράση : είμαι μια έξυπνη χελώνα		
Να ακούσετε τη φράση: Game over		

3^η Διδακτική ώρα

Εντολές εισόδου δεδομένων

ΣΤΟΧΟΙ

- Στόχοι του μαθήματος είναι οι μαθητές να:
- εισάγουν δεδομένα μέσω πλήκτρων και κουμπιών στην οθόνη
- εισάγουν κείμενο και ομιλία

Εισαγωγή

Οι εντολές εισόδου δεδομένων στο περιβάλλον του PencilCode υλοποιούνται με συναρτήσεις.

Υπάρχουν δύο βασικές εντολές εισόδου δεδομένων η `read` και η `readnum`

Η `read` διαβάζει ένα αλφαριθμητικό από το πληκτρολόγιο και η σύνταξη της είναι

```
read 'Κείμενο επεξήγησης', (Μεταβλητή) -> έξοδος συνάρτησης
```

Το κείμενο που θα εισάγουμε θα αποθηκευτεί στη μεταβλητή

Π.χ

```
read 'Πως σε λένε;', (x) ->
  write 'Γεια σου' + x
```

Η `readnum` διαβάζει ένα αριθμό από το πληκτρολόγιο και η σύνταξή της είναι

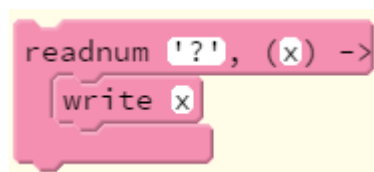
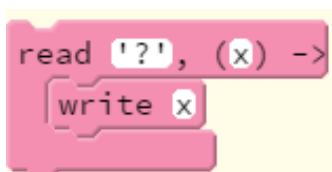
```
readnum 'Κείμενο επεξήγησης', (Μεταβλητή) -> έξοδος συνάρτησης
```

Ο αριθμός που θα εισάγουμε θα αποθηκευτεί στη μεταβλητή

Π.χ

```
readnum 'Your age?', (n) ->
  write 'Next year you will be ' + (n + 1)
```

Τα αντίστοιχα πλακίδια των εντολών `read` και `readnum` βρίσκονται στην ομάδα εντολών `Text`



Εκτός από κείμενο και αριθμούς υπάρχει η δυνατότητα εισαγωγής δεδομένων από το μικρόφωνο (ηχητικά).

Η αντίστοιχη εντολή είναι η `listen` με σύνταξη

```
listen 'Κείμενο", (Μεταβλητή) -> έξοδος συνάρτησης
```

Η πρόταση που θα πούμε θα αποθηκευτεί στη μεταβλητή

Π.χ

```
listen 'Say something', (t) ->
  say 'You said: ' + t
```

Εκτός από τις εντολές αυτές υπάρχουν διαθέσιμες εντολές (συναρτήσεις) οι οποίες ενεργοποιούνται μόλις γίνει μια συγκεκριμένη ενέργεια είτε από το ποντίκι είτε από το πληκτρολόγιο

Μια τέτοια εντολή είναι η `click` η οποία ενεργοποιείται μόλις πατηθεί το πλήκτρο του ποντικιού.

Η σύνταξή της είναι

```
click () -> εντολές
```

Π.χ

```
click () ->  
  say 'hello'
```

Αντίστοιχη εντολή που ενεργοποιείται από το πληκτρολόγιο είναι η `keydown` η οποία ελέγχει αν έχει πατηθεί κάποιο πλήκτρο.

Η σύνταξή της είναι

```
keydown (Πλήκτρο) -> εντολές
```

Π.χ

```
keydown 'A', ->  
  tone 'C'
```

Τέλος μπορούμε να δημιουργήσουμε ένα κουμπί στην οθόνη το πάτημα του οποίου θα εκτελεί μια εργασία. Η εντολή δημιουργία του κουμπιού είναι η `button`.

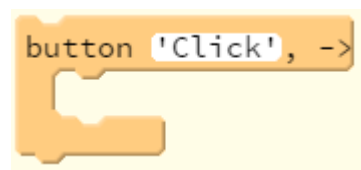
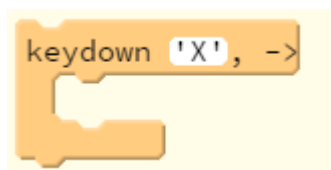
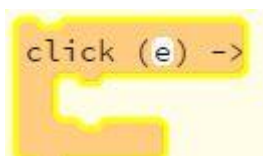
Η σύνταξή της είναι

```
button (Ταμπέλα κουμπιού) -> εντολές
```

Π.χ

```
button 'good morning', ->  
  say 'good morning'
```

Τα αντίστοιχα πλακίδια των εντολών `click`, `keydown` και `button` βρίσκονται στην ομάδα εντολών `Control`



Σχέδιο μαθήματος

Ο Διδάσκων δείχνει παραδείγματα στους μαθητές για τη χρήση των εντολών `read` και `readnum`, `click`, `button` και `keydown`. Στο σημείο αυτό δεν πρέπει να γίνει αναφορά στην έννοια των συναρτήσεων αλλά να διδαχθούν ως σύνταξη της εντολής. Γίνεται αναφορά στο γεγονός ότι η απάντηση στην εντολή `read` και `readnum`, αποθηκεύεται προσωρινά σε μια μεταβλητή.

Στη συνέχεια μοιράζεται το φύλλο εργασίας όπου οι μαθητές καλούνται να δουν άλλα παραδείγματα των παραπάνω εντολών, εφαρμόζοντας τα μόνοι τους στο pencilcode, και στη συνέχεια να φτιάξουν κάποια δικά τους.

Σχολείο

Ημερομηνία.....

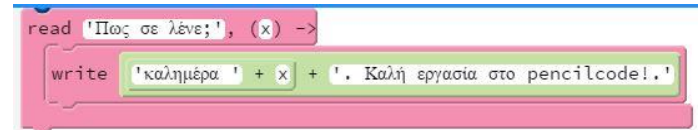
Όνοματεπώνυμο..... Τμήμα..... Αριθ. Καταλ.....

Φύλλο εργασίας - Είσοδος και έξοδος δεδομένων

Στις εικόνες βλέπετε τις εντολές με πλακίδια και στη συνέχεια δίνονται με κώδικα .

☞ Δοκιμάστε τις επόμενες εντολές

ΠΡΟΣΟΧΗ! Προσέχουμε η εντολή `write` να γραφτεί πιο μέσα από την εντολή `read` στο περιβάλλον με τον κώδικα είτε με 2 κενά είτε με ένα `tab`



```
read 'Πως σε λένε;', (x) ->
write 'καλημέρα ' + x + '. Καλή εργασία στο pencilcode!.'
```

```
read 'Πως σε λένε;', (x) ->
  write 'καλημέρα ' + x + '. Καλή εργασία στο pencilcode!.'
```

☞ Δώσε την απάντησή σου και πάτα Submit

Πως σε λένε:

Χρησιμοποίησες πλακίδια (Π) ή κώδικα (Κ) ;.....

☞ Τροποποίησε τις παραπάνω εντολές και φτιάξτε τη δική σου ερώτηση.



```
readnum 'Πόσο χρονών είσαι', (n) ->
say 'you'+ 'are'+ n + 'years'+ ' old'
```

```
readnum 'Πόσο χρονών είσαι', (n) ->
  say 'you'+ 'are'+ n + 'years'+ ' old'
```

☞ Δώσε την απάντησή σου και πάτα Submit

☞ Χρησιμοποίησες πλακίδια (Π) ή κώδικα (Κ) ;.....

☞ Τροποποίησε τις παραπάνω εντολές και φτιάξτε τη δική σου ερώτηση

☞ Δοκίμασε τις εντολές

```
click (e) ->
  tone 'B', 2
```

```
keydown 'A', ->
  write 'πάτησες το πληκτρο A'
```

```
button 'Good morning', ->
  say 'good morning'
```

4^η – 5^η Διδακτική ώρα

Εντολές σχεδίασης

ΣΤΟΧΟΙ

Στόχος του μαθήματος είναι οι μαθητές να:

- Δημιουργούν απλά σχέδια δίνοντας εντολές κίνησης, σε μια χελώνα, είτε χρησιμοποιώντας τη γεωμετρία της χελώνας είτε χρησιμοποιώντας καρτεσιανές συντεταγμένες.

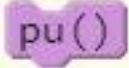
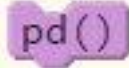
Βασικές εντολές

Εντολές κίνησης - στροφής

	<code>fd 100</code>	Κίνηση Μπροστά 100 εικονοστοιχεία
	<code>rt 90</code>	Δεξιά στροφή 90 μοίρες
	<code>lt 90</code>	Αριστερή στροφή 90 μοίρες
	<code>bk 100</code>	Κίνηση πίσω 100 εικονοστοιχεία
	<code>home()</code>	Επιστροφή της χελώνας στην αρχική της θέση
	<code>rt 180, 100</code> <code>lt 180, 100</code>	Δημιουργία ενός τόξου 180 μοιρών με ακτίνα (κύκλου) 100 εικονοστοιχεία rt από αριστερά προς δεξιά lt από δεξιά προς αριστερά
	<code>moveto 100, 50</code> <code>movexy 30, 20</code>	Η <code>moveto</code> σε μεταφέρει στις οριζόμενες συντεταγμένες x, y ενώ η <code>movexy</code> προκαλεί μετατόπιση x, y σε σχέση με το σημείο που βρίσκεται

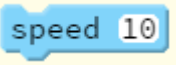
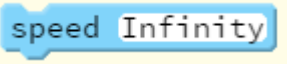
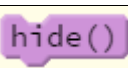
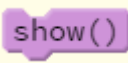
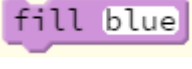

Στα πλακίδια art (μωβ)

	<code>dot green, 50</code> 	Σχεδιάζει ένα κύκλο με συγκεκριμένο χρώμα (πράσινο) και διάμετρο (50) στη θέση που βρίσκεται η χελώνα
	<code>box red, 50</code> 	Σχεδιάζει ένα τετράγωνο με συγκεκριμένο χρώμα (κίτρινο) και πλευρά (50) στη θέση που βρίσκεται η χελώνα
	<code>pen purple, 10</code>	Ορίζει το χρώμα και το μέγεθος του στυλό που θα χρησιμοποιήσει η χελώνα στις επόμενες κινήσεις της

	pu() ή pen up ή pen off	Η χελώνα κατά τη μετακίνησή της δεν αφήνει ίχνος. Όμοια η εντολή pen up
	pd() ή pen on	Επαναφέρει την χελώνα στο να αφήνει ίχνος κατά τη μετακίνησή της

Βοηθητικό

ς εντολές

	speed 10	Αύξηση ταχύτητας σχεδίασης σε 10 εντολές ανά δευτερόλεπτο
	Speed Infinity	Άμεση εκτέλεση όλων των εντολών (βλέπουμε το τελικό αποτέλεσμα)
	hide()	Κρύβει τη χελώνα από την οθόνη
	show()	Εμφανίζει ξανά τη χελώνα
	fill blue	Γεμίζει την προηγούμενη περιοχή με το αναφερόμενο χρώμα
	sc()	Clear screen Καθαρίζει την περιοχή σχεδίασης.

Σχέδιο μαθήματος για 4^η και 5^η διδακτική ώρα

4^η διδακτική ώρα

Προτείνεται να μη μοιραστεί φύλλο εργασίας, αλλά οι επόμενες δραστηριότητες να ζητηθούν από τον διδάσκων γράφοντας τες στον πίνακα είτε με προβολή.


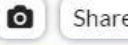


Λέμε κάθε φορά σε ποια ομάδα εντολών πλακιδίων θα τις βρουν. Καλό είναι επίσης να εξηγούμε τη σημασία των εντολών, καθώς και τη μετάφραση τους στα ελληνικά.

Δραστηριότητες

-Ζητάμε από τους μαθητές να δοκιμάσουν τις εντολές

```
dot black, 100
dot orange, 50
```

-Τους λέμε ότι ο αριθμός δηλώνει την ακτίνα του κύκλου με μονάδα μέτρησης το εικονοστοιχείο και τους ζητάμε να δοκιμάσουν δικές τους τιμές διαφορετικά χρώματα. Τους δείχνουμε που μπορούν να βρουν περισσότερα διαθέσιμα :

τα διαθέσιμα χρώματα μπορείτε να τα βρείτε αν επιλέξετε Guide	     
---	---

και στη συνέχεια

[Quick Reference Sheet](http://pencilcode.net/material/reference.pdf)

<http://pencilcode.net/material/reference.pdf>

Ή κατευθείαν πάνω στην διπλανή
εικόνα



-Ζητάμε από τους μαθητές να δοκιμάσουν τις εντολές την επόμενη ομάδα εντολών:

```
dot pink, 70  
fd 100  
box gold, 60  
dot midnightblue, 25  
hide()
```

-Ζητάμε από τους μαθητές να μας πουν τι νομίζουν ότι κάνουν οι εντολές νέες εντολές που συνάντησαν παραπάνω

-Ας δοκιμάσουν τώρα την εντολή

```
cs()
```

Καλό είναι να χρησιμοποιούμε αυτή την εντολή ανάμεσα στις δοκιμές

-Ζητάμε τώρα να δοκιμάσουν τις εξής εντολές:

```
show()  
home()  
pen blue, 10  
fd 100  
rt 90  
bk 150  
lt 90  
fd 50  
pu()  
fd 50  
dot red, 40
```

-Συζητάμε με τους μαθητές τι κάνουν οι παραπάνω εντολές.

-Ζητάμε από τους μαθητές να φτιάξουν ένα τετράγωνο με μήκος πλευράς 100 pixels, πάχος γραμμής 5 και με διαφορετικό χρώμα στην κάθε πλευρά.

Ας δοκιμάσουν τις επόμενες εντολές

```
click (e) ->  
moveto e.x, e.y
```

Εξηγούμε ότι το x και y είναι καρτεσιανές συντεταγμένες. Στη συνέχεια πείτε στους μαθητές να κάνουν "κλικ" σε διάφορα σημεία του χώρου σχεδίασης.

-Ζητάμε να δοκιμάσουν τις εντολές

```
pen olive, 5  
rt 180, 50
```

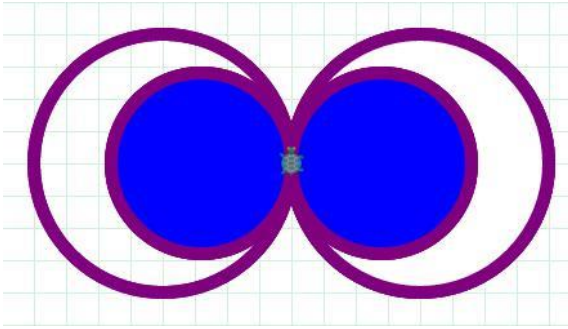
-Ζητούμε να φτιάξουν ένα κύκλο κύκλο ακτίνας 100 εικονοστοιχείων

-Αφού φτιάξουν τον κύκλο, χωρίς να τον σβήσουν να δώσουν την εντολή

```
fill red
```

-Ζητούμε από τους μαθητές να φτιάξουν το παρακάτω σχήμα:

-μπορεί να υλοποιηθεί με διαφορετικούς τρόπους, το συζητούμε αυτό στην τάξη και κάθε τρόπο που βλέπουμε να χρησιμοποιείτε ζητούμε από τον μαθητή να καταγράψει τις εντολές που χρησιμοποίησε στον πίνακα, με σωστή σύνταξη, όταν η πλειοψηφία της τάξης ολοκληρώσει τη δραστηριότητα.



Προτείνεται όσοι μαθητές ολοκληρώσουν τις προηγούμενες δραστηριότητες, να τους ζητηθεί να φτιάξουν ένα δικό τους πρωτότυπο σχήμα

5^η διδακτική ώρα

Μοιράζεται το επόμενο φύλλο εργασίας.

Επισημαίνεται η αρχική και η τελική θέση της χελώνας. Επίσης ο διδάσκων υπενθυμίζει στους μαθητές στοιχεία από τη γεωμετρία για τη σχεδίαση του 2^{ου} σχήματος. Να επισημανουμε ότι η

Σημείωση: Στην περίπτωση που οι μαθητές έχουν εξοικειωθεί με τη χρήση της χελώνας σε κάποιο άλλο logo like περιβάλλον, η διδακτική ώρα μπορεί να παραληφθεί.

Σχολείο

Ημερομηνία.....

Όνοματεπώνυμο.....

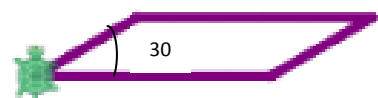
Τμήμα..... Αριθ. Καταλ.....

Εντολές σχεδίασης

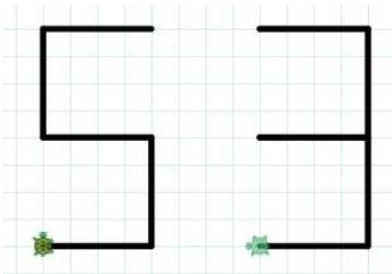
- 1) Να δώσεις τις εντολές για το σχηματισμό του παρακάτω ορθογωνίου παραλληλογράμμου με πλευρές 50 και 100 pixels. **Ποιο περιβάλλον χρησιμοποίησες?**



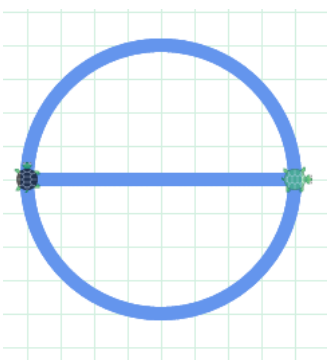
- 2) Να δώσεις τις εντολές για το σχηματισμό του παρακάτω παραλληλογράμμου με πλευρές 100 και 200 pixels. **Ποιο περιβάλλον χρησιμοποίησες?**



- 3) Γράψτε τις εντολές προκειμένου η χελώνα να ζωγραφίσει, τον ακόλουθο αριθμό, όπως φαίνεται. Η χελώνα θα ξεκινήσει από κάτω μέρος του αριθμού 5 και θα τελειώσει στο κάτω μέρος του αριθμού 3, όπως φαίνεται στο σχήμα. Τα ευθύγραμμα τμήματα έχουν μήκος 50 pixels. **Ποιο περιβάλλον χρησιμοποίησες?**



- 4) Γράψτε τις εντολές προκειμένου η χελώνα να ζωγραφίσει το ακόλουθο σχήμα, όπου φαίνεται η αφετηρία της χελώνας και το τέλος. Η ακτίνα του κύκλου είναι 50 pixels. **Ποιο περιβάλλον χρησιμοποίησες?**



6η διδακτική ώρα

Δομή Επανάληψης: Η εντολή for

ΣΤΟΧΟΙ

Στόχος του μαθήματος είναι οι μαθητές να μπορούν:

- να ξεχωρίζουν πότε μπορούν να χρησιμοποιήσουν δομή επανάληψης
- να χρησιμοποιούν την εντολή `for`

Εισαγωγή

Η επανάληψη επιτρέπει σε ένα πρόγραμμα να εκτελέσει ένα πλήθος εντολών πολλές φορές χωρίς να χρειάζεται να αναφέρει αναλυτικά.

Η επανάληψη στο PencilCode γίνεται με τρεις τρόπους από τους οποίους εδώ θα παρουσιαστεί η δομή `for`.

Η δομή `for` επαναλαμβάνει μια ομάδα εντολών για ένα σταθερό αριθμό επαναλήψεων οι οποίες δηλώνονται σε μια λίστα.

Το πλακίδιο `for` βρίσκεται στην ομάδα πλακιδίων Control.

Για παράδειγμα η επόμενη δομή `for` θα γράψει τη λέξη "Hello" τρεις φορές



Σχέδιο μαθήματος

Μοιράζεται το φύλλο εργασίας. Ο διδάσκων λέει στους μαθητές ότι η δομή επανάληψης `for` βρίσκεται στα πλακίδια στην ομάδα εντολών Control –ελέγχου με χρώμα πορτοκαλί ανοιχτό.

Ζητείται από τους μαθητές, να κάνουν την πρώτη δραστηριότητα χρησιμοποιώντας τα πλακίδια.

ο Διδάσκων συζητά με τους μαθητές τη χρήση της εντολής `for`.

Τους ζητά να δουν την εντολή στο περιβάλλον κειμένου και να προσέξουν τη σύνταξη της.

Είναι σημαντικό να αντιληφθούν ότι οι εντολές που είναι μέσα στο βρόχο και επαναλαμβάνονται, γράφονται με μια εσοχή στην αρχή της γραμμής, δηλαδή όχι στην ίδια "κάθετο" με την εντολή `for`. Η εσοχή αυτή δημιουργείται είτε πατώντας 2 κενά είτε το πλήκτρο `tab`.

Σχολείο

Ημερομηνία.....

Όνοματεπώνυμο.....

Τμήμα..... Αριθ. Καταλ.....

Δομή επανάληψης - η εντολή for

1) Δοκιμάστε τις παρακάτω εντολές χρησιμοποιώντας τα πλακίδια και στη συνέχεια δείτε τις ως κώδικα. Προσέξτε τη σύνταξή τους.

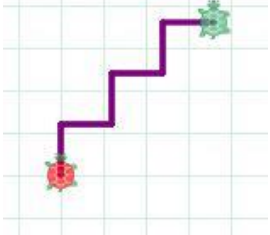
Εντολές- Κώδικας	Πλακίδια	Αποτέλεσμα - output
for [1..5] write 'καλημέρα'		

2) Χρησιμοποιώντας την εντολή **for**, δώστε τις κατάλληλες εντολές ώστε να ακουστεί 4 φορές η φράση GOOD MORNING. **Χρησιμοποιήστες Πλακίδια ή Κώδικα; _____**

Γράψτε τις εντολές στο φύλλο εργασίας, προσέχοντας τη σύνταξη τους.

.....
.....

3) Δώστε κατάλληλες εντολές για να φτιαχτεί το παρακάτω σχήμα, το ύψος και το πλάτος σε κάθε σκαλί είναι 30 εικονοστοιχεία. Η χελώνα ξεκινάει από κάτω και τελειώνει πάνω, όπως φαίνεται στην εικόνα. Γράψτε τις εντολές στο φύλλο εργασίας, προσέχοντας τη σύνταξή τους.

	Γράψτε τις εντολές	Χρησιμο- ποίησες Πλακίδια ή Κώδικα;
α) αναλυτικά οι εντολές χωρίς τη χρήση της εντολής for	
β) με την εντολή for	

7η διδακτική ώρα

Η εντολή for - Κανονικά πολύγωνα

ΣΤΟΧΟΙ

Στόχος του μαθήματος είναι οι μαθητές να μπορούν:

- να χρησιμοποιούν την εντολή for για τη δημιουργία κανονικών πολυγώνων
- να αναγνωρίζουν ότι η χελώνα στρίβει την εξωτερική γωνία του πολυγώνου
- να υπολογίζουν την εξωτερική γωνία του κανονικού πολυγώνου από το πλήθος των κορυφών του

Σχέδιο μαθήματος

Μοιράζεται το Φύλλο εργασίας στους μαθητές.

Ο Διδάσκων ζητά από τους μαθητές να κάνουν τη πρώτη δραστηριότητα. Με αφορμή το αποτέλεσμα της πρώτης δραστηριότητας, γίνεται συζήτηση με τους μαθητές, για την τιμή της στροφής, με τη μαιευτική μέθοδο.

Οι μαθητές καλούνται να συμπληρώσουν το υπόλοιπο φύλλο εργασίας.

Σχολείο

Ημερομηνία.....

Όνοματεπώνυμο.....

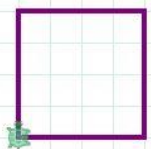
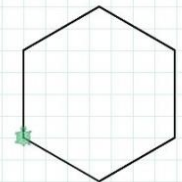
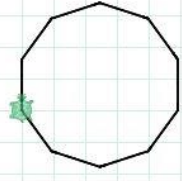
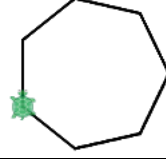
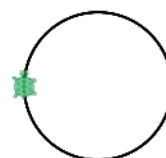
Τμήμα..... Αριθ. Καταλ.....

Σχεδίαση - Δομή επανάληψης- Κανονικά Πολύγωνα

1) Δοκιμάστε τις παρακάτω εντολές. Τι σχήμα προκύπτει; **Τι παρατηρείτε;**

Εντολές- Κώδικας	Πλακίδια	Σχήμα
<pre>pen black for [1..3] fd 100 rt 120</pre>		

3) Δώστε τις κατάλληλες εντολές για τη δημιουργία των παρακάτω σχημάτων, **χρησιμοποιώντας την εντολή for** (η χελώνα αρχίζει και τελειώνει στην ίδια θέση που φαίνεται). Γράψτε τις εντολές στο Φύλλο εργασίας, προσέχοντας τη σύνταξή τους.

		εντολές	Χρησιμοποίησες Πλακίδια ή Κώδικα;
τετράγωνο			
Κανονικό εξάγωνο			
Κανονικό δεκάγωνο			
Κανονικό επτάγωνο			
Κύκλος (προσεγγιστικά ως ένα κανονικό πολύγωνο με πολλές γωνίες)			

8^η – 10^η διδακτική ώρα

Υποπρογράμματα –συναρτήσεις

ΣΤΟΧΟΙ

Στόχος του μαθήματος είναι οι μαθητές να μπορούν:

- Να δημιουργούν συναρτήσεις
- Να δημιουργούν προγράμματα τα οποία καλούν συναρτήσεις

Εισαγωγή

Μια συνάρτηση είναι ένα ανεξάρτητο τμήμα κώδικα που αποτελεί ένα υποπρόγραμμα, δηλαδή μπορεί να αποτελέσει τμήμα ενός άλλου προγράμματος. Οι συναρτήσεις χρησιμοποιούνται από τους προγραμματιστές για να διαιρέσουν μεγάλα προγράμματα σε ένα πλήθος από μικρότερα. Ορισμένα από τα πλεονεκτήματα των συναρτήσεων είναι:


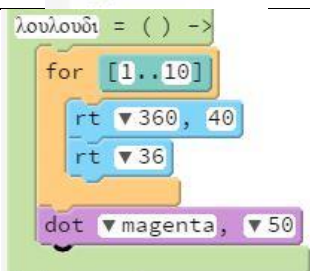
Οι συναρτήσεις επιτρέπουν την επαναχρησιμοποίηση του κώδικα. Αν δημιουργήσουμε μια συνάρτηση, ο κώδικάς της μπορεί να χρησιμοποιηθεί σε πολλά σημεία του προγράμματος χωρίς να χρειάζεται να ξαναγράψουμε τις εντολές.

Οι συναρτήσεις επιτρέπουν τον έλεγχο της εκτέλεσης του κώδικα. Η διάσπαση ενός μεγάλου μήκους κώδικα σε μικρότερα επιτρέπει τον καλύτερο έλεγχο των εντολών.

Προσοχή: Οι εντολές που βρίσκονται σε μια συνάρτηση δεν εκτελούνται αμέσως με την ολοκλήρωση της συνάρτησης αλλά αργότερα όταν κληθεί η συνάρτηση αυτή.


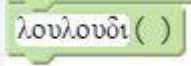
Δημιουργία συνάρτησης (χωρίς μεταβλητή)

Στην ομάδα πλακιδίων Operators υπάρχει το αντίστοιχο πλακίδιο ορισμού μιας συνάρτησης. Σε κώδικα μια συνάρτηση ορίζεται από το όνομα της συνάρτησης το χαρακτήρα " = ", δύο παρενθέσεις "()" μέσα στις οποίες τοποθετούνται οι μεταβλητές της συνάρτησης και τέλος οι χαρακτήρες "-->". Μετά το "βέλος" ακολουθούν οι εντολές που περιλαμβάνει η συνάρτηση αν οι εντολές είναι περισσότερες από μια τότε γράφονται σε νέα γραμμή με εσοχή δύο χαρακτήρων στην αρχή της.

	Το πλακίδιο ορισμού συνάρτησης
	<pre>λουλουδι = () -> pen purple, 5 for [1..10] rt 360, 40 rt 36 dot magenta, 50</pre>

Οι παρενθέσεις, μπορούν να παραλειφθούν, αν δώσουμε τις εντολές με κώδικα.

Πως την καλούμε:

	
	λουλουδι ()

Σχέδιο μαθήματος

Τα υποπρογράμματα και η χρήση τους είναι μια απαραίτητη βασική γνώση και δεξιότητα που πρέπει να αποκτήσουν οι μαθητές. Στο περιβάλλον του Pencil Code τα υποπρογράμματα υλοποιούνται με συναρτήσεις.

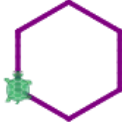

Πριν οι μαθητές αρχίσουν να εργάζονται, ο διδάσκων δείχνει ένα παράδειγμα για τη δημιουργία υποπρογράμματος στον υπολογιστή του, πρώτα χρησιμοποιώντας πλακίδια και στη συνέχεια τους δείχνει τον κώδικα και επισημαίνει τη σύνταξη: Οι εντολές που περιλαμβάνονται στο υποπρόγραμμα γράφονται είτε με 2 κενά στην αρχή κάθε γραμμής εντολής είτε με ένα πάτημα του πλήκτρου tab στην αρχή.

Επισημαίνει στους μαθητές ότι όταν δημιουργούν ένα υποπρόγραμμα, δεν εκτελούνται οι εντολές που περιέχει, παρά μόνο αποθηκεύονται και με το όνομα της συνάρτησης μπορούμε να τις καλέσουμε και αν εκτελεστούν.


Στη συνέχεια ο διδάσκων καλεί τη συνάρτηση, χρησιμοποιώντας πλακίδια. Επισημαίνει στους μαθητές τα διαφορετικά πλακίδια για τη δημιουργία και τη χρήση συνάρτησης.

Στη συνέχεια ο διδάσκων με τη χρήση των κατωτέρω παραδειγμάτων και με χρήση πλακιδίων αλλά και με κώδικα δημιουργεί πρώτα τη συνάρτηση, "προσπαθεί" να την εκτελέσει για να δείξει στους μαθητές ότι οι εντολές μια συνάρτησης εκτελούνται μόνο όταν την καλέσεις. Στη συνέχεια κάνει διαδοχικές κλήσεις της συνάρτησης για την δημιουργία του επιθυμητού σχήματος

Παράδειγμα 1

<pre>εξάγωνο = () -> pen purple, 3 for [1..6] fd 40 rt 60</pre>	Ο κώδικας δε βγάζει αποτέλεσμα
εξάγωνο()	
<pre>for [1..4] εξάγωνο() rt 90</pre>	

Παράδειγμα2

<pre>circle1=()-> pen black, 10 rt 360, 70 fill red circle2 = ()-> pen green, 6 rt 360, 50 fill yellow circle3 = ()-> pen purple, 3 rt 360, 30 fill cyan</pre>	<p>Ο κώδικας δε βγάζει αποτέλεσμα</p>
<pre>speed 20 circle1() jumpxy 0, 120 circle2() jumpxy 0, 80 circle3()</pre>	

Μοιράζεται το 1^ο φύλλο εργασίας στους μαθητές. Επισημαίνεται στους μαθητές να είναι προσεκτικοί στην ανάγνωση των οδηγιών του φύλλου εργασίας.

Μπορούν να διατεθούν άλλες 1-2 διδακτικές ώρες με τα επόμενα 2 φύλλα εργασίας.

Σχολείο

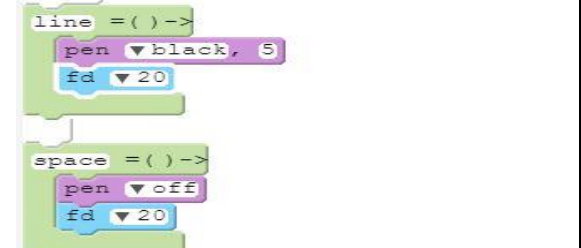
Ημερομηνία.....

Όνοματεπώνυμο.....

Τμήμα..... Αριθ. Καταλ.....


Συναρτήσεις

☞ Δώστε τις παρακάτω εντολές που θα δημιουργήσουν τις συναρτήσεις line και space. Τα πλακίδια των συναρτήσεων που θα χρειαστείτε είναι στην ομάδα Operators, με πράσινο χρώμα

 <pre> line = () -> pen ▼ black, 5 fd ▼ 20 space = () -> pen ▼ off fd ▼ 20 </pre>	<pre> line = () -> pen black, 5 fd 20 space = () -> pen off fd 20 </pre>
---	---

Μόλις δημιουργήσατε τις πρώτες σας συναρτήσεις. Προέκυψε κάποιο αποτέλεσμα; ΝΑΙ ή ΟΧΙ (κύκλωσε)


☞ **Χωρίς να σβήσετε** τις παραπάνω εντολές, δοκιμάστε τις εξής . Τι παρατηρείτε;

 <pre> line () space () </pre>	<pre> line () space () </pre>
---	-------------------------------

☞ **Σβήστε τις τελευταίες 2 εντολές, αλλά ΜΗ να σβήσετε τις συναρτήσεις line και space.**

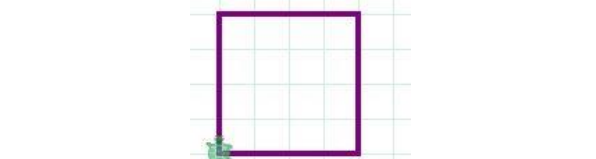
Χρησιμοποιείτε τις εντολές line(), space() και for και όποια άλλη εντολή χρειάζεται και δημιουργήστε το παρακάτω σχήμα. Σημειώστε τις εντολές στο φύλλο εργασίας, προσέχοντας τη σύνταξή τους.

Χρησιμοποιήσατε πλακίδια ή κώδικα;.....

	<p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p>
---	---

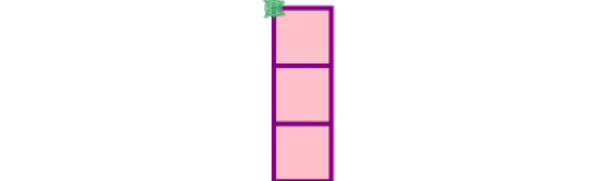
☞ Δημιουργήστε μια συνάρτηση με το όνομα tetragono για τη δημιουργία ενός τετραγώνου και καλέστε την ώστε να το δημιουργήσει. Προσοχή η χελώνα θα καταλήγει στην αρχική της θέση. Χρησιμοποιείτε την εντολή for. Σημειώστε τις εντολές στο φύλλο εργασίας, προσέχοντας τη σύνταξή τους.

Χρησιμοποιήσατε πλακίδια ή κώδικα;.....

	<p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p>
---	---

☞ Χρησιμοποιείτε τη συνάρτηση tetragono() σε συνδυασμό με την εντολή for και όποιες άλλες εντολές χρειάζονται για να δημιουργήσετε το παρακάτω σχήμα. Σημειώστε τις εντολές στο φύλλο εργασίας, προσέχοντας τη σύνταξή τους.

Χρησιμοποιήσατε πλακίδια ή κώδικα;.....

	<p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p>
---	---

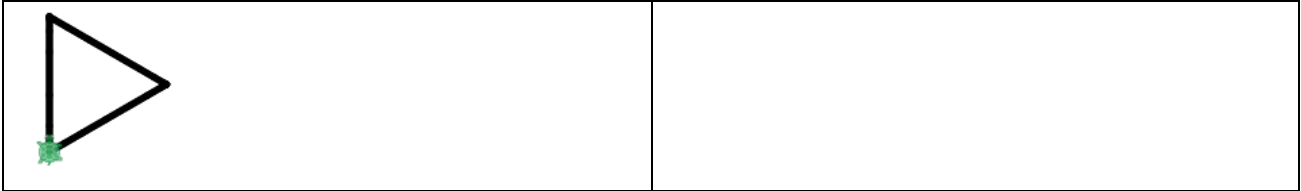
Σχολείο

Ημερομηνία.....

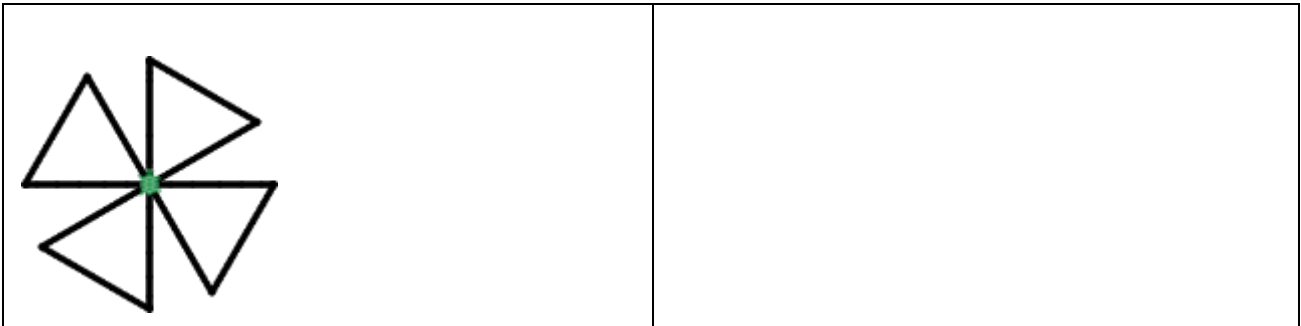
Όνοματεπώνυμο..... Τμήμα..... Αριθ. Καταλ.....

Συναρτήσεις-σχήματα εκ περιστροφής

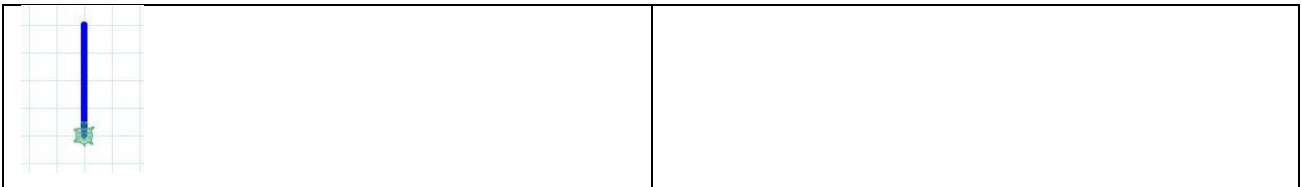
- ☞ Δημιουργήστε συνάρτηση με το όνομα `isopleuro` για τη δημιουργία ισόπλευρου τριγώνου, με πλευρές μήκους 60 εικονοστοιχείων, πάχους γραμμής 5, και χρώματος μαύρο. Κάλεστε την ώστε αν δημιουργηθεί το σχήμα που βλέπετε **Χρησιμοποιήσατε πλακίδια ή κώδικα;**



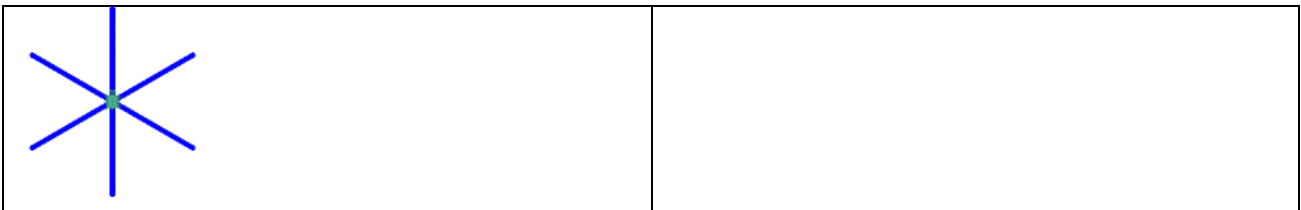
- ☞ Στη συνέχεια χρησιμοποιώντας τη συνάρτηση `isopleuro` σε συνδυασμό με την εντολή `for` και άλλες εντολές δημιουργήστε το παρακάτω σχήμα . **Χρησιμοποιήσατε πλακίδια ή κώδικα;**



- ☞ Δημιουργήστε συνάρτηση με το όνομα `aktina` για τη δημιουργία ευθυγράμμου τμήματος (ακτίνας) μήκους 50 εικονοστοιχείων, χρώματος μπλε και πάχους γραμμής 3 (Προσοχή: Η χελώνα αρχίζει και τελειώνει στην ίδια θέση). **Χρησιμοποιήσατε πλακίδια ή κώδικα;**



- ☞ Χρησιμοποιώντας τη συνάρτηση `aktina` , την εντολή `for` και όποιες άλλες εντολές νομίζετε, φτιάξτε το επόμενο σχήμα. **Χρησιμοποιήσατε πλακίδια ή κώδικα;....**



Σχολείο

Ημερομηνία.....

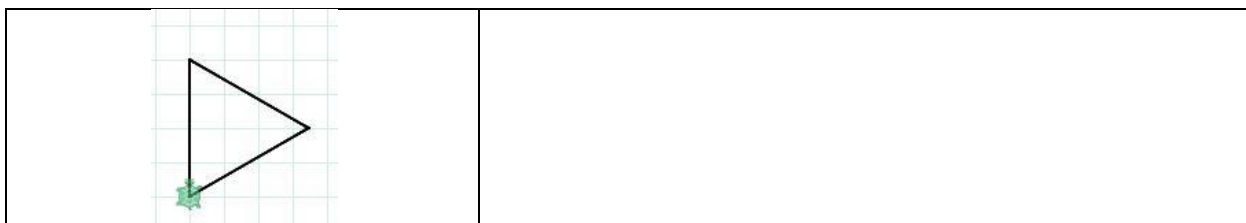
Όνοματεπώνυμο..... Τμήμα..... Αριθ. Καταλ.....

Υπερδιαδικασίες- Σπίτι

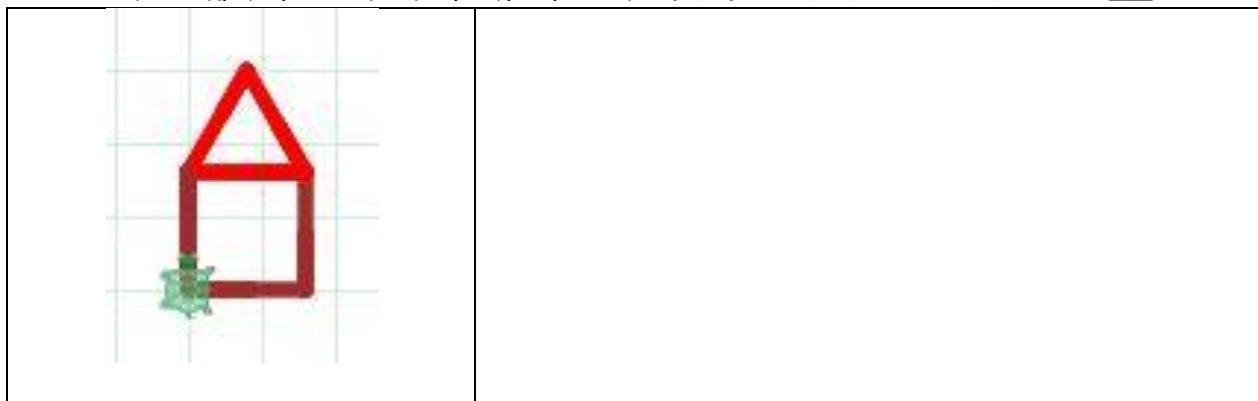
- ☞ Φτιάξε μια συνάρτηση **τετράγωνο** για τη δημιουργία τετραγώνου, πλευράς 100 εικονοστοιχείων.
Χρησιμοποίησες Πλακίδια ή Κώδικα;___



- ☞ Φτιάξε μια συνάρτηση με το όνομα **ισόπλευρο** (ΜΗ ΣΒΗΣΕΙΣ ΤΗ ΣΥΝΑΡΤΗΣΗ τετράγωνο) και στη συνέχεια κάλεσέ τη για να δημιουργήσεις το εξής **ισόπλευρο** τρίγωνο με μήκος πλευράς 100 εικονοστοιχείων.
Χρησιμοποίησες Πλακίδια ή Κώδικα;___



- ☞ Χρησιμοποιώντας τις συναρτήσεις **τετράγωνο** και **ισόπλευρο** και όποιες άλλες εντολές χρειάζονται δημιουργήστε το παρακάτω σπίτι. Φρόντισε η χελώνα να αρχίσει και να τελειώσει στην θέση που φαίνεται. Το τετράγωνο κάντο καφέ και τη σκεπή κόκκινη. Αν θες μπορείς να γεμίσεις τη σκεπή με κόκκινο χρώμα.
- ☞ Φτιάξε μια συνάρτηση με το όνομα **σπίτι**, η οποία θα περιλαμβάνει τις συναρτήσεις και τις εντολές που χρησιμοποίησες προηγουμένως. **Χρησιμοποίησες Πλακίδια ή Κώδικα;___**



- ☞ Φτιάξε μια συνάρτηση με το όνομα **χωριό**, που θα φτιάχνει 4 σπίτια, το ένα δίπλα στο άλλο.
Χρησιμοποίησες Πλακίδια ή Κώδικα;___

11^η - 15^η διδακτική ώρα

Οι μεταβλητές κι η χρήση τους

Η μεταβλητή είναι ένα όνομα για μια τιμή που θέλουμε να αποθηκεύσουμε για να την χρησιμοποιήσουμε αργότερα.

I. Δημιουργώ μεταβλητές – δίνω τιμές σε αυτές

Δημιουργούμε μια μεταβλητή και αποδίδουμε τιμή σε αυτήν με το =

Παραδείγματα

<pre>x= 2.5 write x*x</pre>
<pre>onoma='Maria' write onoma say onoma write 'onoma'</pre>
<pre>message = 'καλώς ήρθατε.' see 'message' see message</pre>
<pre>x = 20 box black, x * 5 box white, x * 4 box black, x * 3 box white, x * 2</pre>

Στον προγραμματισμό η εκχώρηση $x = \text{τιμή}$ αντιπροσωπεύει μια αποθήκευση στη μνήμη με όνομα x . Στα μαθηματικά το $y = x^2$ ερμηνεύεται ως ένας ορισμός.

Συνηθισμένες παρανοήσεις :

<pre>x=5 y=4 x=y</pre>	Η τελευταία εντολή αποδίδει στο x την τιμή του y στα μαθηματικά αυτό είναι λάθος
<pre>x=(random 6) write = x + x</pre>	Ο μαθητής μπορεί να θεωρήσει ότι το x ορίζεται σε κάθε εμφάνισή του να είναι τυχαίος αριθμός, οπότε να περιμένει ότι θα εμφανιστεί ένα αποτέλεσμα όπως $4+5=9$. Αυτό όμως δεν ισχύει μέσα στην ίδια εντολή
<pre>write x + 6 x = 5</pre>	Η απόδοση τιμής σε μια μεταβλητή θα πρέπει να προηγείται από τη χρήση της μεταβλητής. Θα πρέπει να τεθεί πρώτα η εντολή $x = 5$ και μετά η $write x + 6$

II. Συναρτήσεις με μεταβλητές (χρήση παραμέτρων)

Οι μεταβλητές που χρησιμοποιούμε στον ορισμό μιας συνάρτησης ονομάζονται παράμετροι (για παράδειγμα το x στη συνάρτηση $tetragonox$ είναι παράμετρος), ενώ όταν καλούμε τη συνάρτηση με μια τιμή ή μια μεταβλητή αυτή ονομάζεται όρισμα (για παράδειγμα το 100 στην κλήση $tetragonox(100)$).

Παράδειγμα 1:

$tetragonox(x) \rightarrow$

```
for [1..4]
  fd x
  rt 90
```

την καλούμε ως εξής

```
tetragonox(100)
```

ή

```
tetragonox 100
```

ή

```
x=100
```

```
tetragonox(x)
```

Παράδειγμα 2:

```
τρίγωνο = (color, n, x) ->
```

```
  pen color, n
```

```
  for [1..3]
```

```
    fd x
```

```
    rt 120
```

την καλούμε ως εξής

```
τρίγωνο (red, 4, 100) ή
```

```
τρίγωνο red, 4, 100
```

ή

```
color=red
```

```
n=4
```

```
x=100
```

```
τρίγωνο(color, n, x)
```

Αν έχουμε γράψει και τις 2 παρακάτω συναρτήσεις με τη σειρά που φαίνονται:

```
τρίγωνο= ->
```

```
  for [1..3]
```

```
    fd 400
```

```
    rt 120
```

```
τρίγωνο = (color, n, x) ->
```

```
  pen color, n
```

```
  for [1..n]
```

```
    fd x
```

```
    rt 360/n
```

Αν καλέσουμε τη διαδικασία

```
τρίγωνο()
```

προφανώς θα καλέσει τη 2η! συνάρτηση, ο ορισμός ο της οποίας αντικατέστησε την πρώτη, και στα μεταβλητές θα χρησιμοποιήσει τις default τιμές. Για το color η τιμή αυτή είναι black, για τις άλλες μεταβλητές η τιμή τους θα είναι άγνωστη (πρακτικά μηδέν).

I. Τοπικές μεταβλητές και γενικές μεταβλητές

Μέσα σε ένα πρόγραμμα οι μεταβλητές που ορίζουμε και η τιμή τους ισχύουν για όλο το πρόγραμμα. Δηλαδή είναι γενικές. Αυτό όμως δεν ισχύει όταν το πρόγραμμα περιέχει συναρτήσεις. Τότε οι μεταβλητές που ορίζονται μέσα σε συναρτήσεις είναι τοπικές, δηλαδή ισχύουν μόνο μέσα στη συνάρτηση ενώ δεν αναγνωρίζονται έξω από αυτήν. Οι γενικές μεταβλητές ισχύουν και μέσα στις συναρτήσεις και η τιμή τους μπορεί να τροποποιηθεί μέσα σε αυτές.

Παραδείγματα :

```
x = 5
f = () ->
  x = 10
  write x
f()
write x
```

βγάζει 10 10 η μεταβλητή x αναγνωρίστηκε και τροποποιήθηκε μέσα στη συνάρτηση

```
x = 5
f = (y) ->
  y = 10
  write y
f(x)
write x
βγάζει (φυσιολογικά) 10 5
```

Αν προστεθεί στο τέλος

```
x = 5
f = () ->
  x = 10
  write x
f()
write x
write y
```

τότε θα βγάλει μήνυμα ότι η μεταβλητή y δεν είναι ορισμένη (ορίστηκε μέσα στη συνάρτηση)

Το ίδιο ισχύει αν χρησιμοποιηθεί η ίδια μεταβλητή εσωτερικά και εξωτερικά της συνάρτησης

```
x = 5
f = (x) ->
  x = 10
  write x
f(x)
write x
```

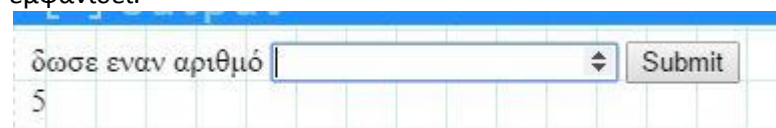
προβλήματα στη χρήση συναρτήσεων

```
x=5
readnum 'δωσε εναν αριθμό', (x) ->
  write x
write x
```

Η τελευταία εντολή write θα εμφανίσει το αποτέλεσμα 5 πριν από την readnum!

Οι συναρτήσεις εκτελούνται παράλληλα με τα προγράμματα ακόμα και αν είναι ενσωματωμένες σε αυτά.

Έτσι εφόσον της έχουμε δώσει μια τιμή της μεταβλητής x προηγουμένως, η τελευταία write θα την εμφανίσει.



δωσε εναν αριθμό Submit

Έστω ότι του δίνουμε τον αριθμό 4. Τότε θα προκύψει το ακόλουθο αποτέλεσμα.

δώσε έναν αριθμό 4

5

4

Αν δεν έχουμε δώσει εξ' αρχής τιμή για τη μεταβλητή x θα εμφανιστεί μήνυμα λάθους.

```
readnum 'δώσε έναν αριθμό', (x) ->  
  write x  
write x
```

Submit
Oops, the computer got confused.
It says: "x is not defined"
Is 'x = something' needed first?
Or are quotes needed around "x"?

Αν θέλουμε η ερώτηση να περιμένει να εισάγει απάντηση ο χρήστης πριν προχωρήσει στην εκτέλεση της επόμενης εντολής

```
x=5  
await readnum 'Δώσε έναν αριθμό', defer x  
write x
```

Δώσε έναν αριθμό Submit

Έστω ότι του δίνουμε τον αριθμό 10. Τότε θα προκύψει το ακόλουθο αποτέλεσμα.

Δώσε έναν αριθμό 10

10

II. Μεταβλητές με τιμές από Λίστα

<pre>for n in [1..5] write 3 * n</pre>	
<pre>for x in [10, 43, 6, 27] write 'ο αριθμός ', x</pre>	
<pre>for x in ['Ελένη', 'Γιάννη', 'Μαρία'] write 'Γεια σου', x</pre>	
<pre>for color in [red, orange, yellow, blue] pen color, 4 fd 100 rt 90</pre>	
<pre>for d in [50, 100, 50, 100] fd d rt 90</pre>	
<pre>for outside in [darkmagenta, violet, pink] for inside in [royalblue, tomato, fuchsia] dot outside, 70 dot inside, 50 fd 100 rt 360/9</pre>	
<pre>mycolors = [red, orange, yellow] for x in mycolors dot x, 100 fd 50</pre>	
<pre>tetragonox = (x)-> for [1..4] fd x</pre>	

<pre> rt 90 pen red for x in [100, 150, 200] tetragonox(x) ή sxima =(times)-> for x in times tetragonox(x) pen red sxima [100, 150, 200] </pre>	
--	--

```

pen purple
for x in [50..1] by -1
  rt 30, x

```

Λίγα στοιχεία για τις λίστες:

Εντολές	Αποτέλεσμα
<pre> onomata = ['Μαρία', ' Ελένη', ' Γιώργος'] </pre>	(Ορισμός μια λίστας)
<pre> write onomata </pre>	Μαρία, Ελένη, Γιώργος
<pre> write onomata.length </pre>	3 (υπάρχουν 3 στοιχεία στη λίστα).
<pre> write onomata[0] </pre>	Μαρία (Το 0 αντιστοιχεί στην πρώτη θέση)
<pre> write onomata[1] </pre>	Ελένη (Το 1 αντιστοιχεί στην δεύτερη θέση)
<pre> write onomata[2] </pre>	Γιώργος
<pre> onomata[0] = 'Σοφία' </pre>	(Απόδοση τιμής στο πρώτο στοιχείο της λίστας)
<pre> write onomata </pre>	Σοφία, Ελένη, Γιώργος

11^η - 12^η ώρα Συναρτήσεις με μεταβλητές

Στόχοι

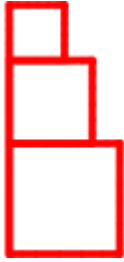
Στόχος του μαθήματος είναι οι μαθητές να μπορούν:

- Να δημιουργούν συναρτήσεις που να περιέχουν παραμέτρους
- Να δημιουργούν προγράμματα τα οποία καλούν συναρτήσεις με χρήση ορισμάτων

Σχέδιο μαθήματος

11^η ώρα:

Ο διδάσκων μοιράζει το φύλλο εργασίας στους μαθητές. Τους ζητά να κοιτάξουν το πρώτο σχήμα:



Τους ζητάει να του πουν πόσα τετράγωνα βλέπουν και πόσες συναρτήσεις για τα τετράγωνα πιστεύουν ότι πρέπει να δημιουργήσουν .

Τους γράφει στον πίνακα το εξής:

```
τετράγωνο = (x) ->  
  for[1..4]  
  fd x  
  rt 90
```

Γίνεται συζήτηση στην τάξη για τη μεταβλητή x.

Με μαιευτική μέθοδο, προκύπτει ποια θα πρέπει να είναι η εντολή για να καλέσουμε τη συνάρτηση για τη δημιουργία τετραγώνου με συγκεκριμένο μήκος πλευράς.

```
τετράγωνο (20) ή  
τετράγωνο 20  
ή  
x=20  
τετράγωνο (x)
```

Στη συνέχεια ο διδάσκων γράφει στον πίνακα

```
τετράγωνο = (color, x, y) ->  
  pen color, y  
  for[1..4]  
  fd x  
  rt 90
```

Γίνεται συζήτηση στην τάξη για τις νέες μεταβλητές

Με μαιευτική μέθοδο, προκύπτει ποια θα πρέπει να είναι η εντολή για να καλέσουμε τη συνάρτηση για τη δημιουργία τετραγώνου με συγκεκριμένο μήκος, χρώματος και πάχους πλευράς.

```
τετράγωνο (red, 100, 5) ή  
τετράγωνο red, 100, 5
```

12^η ώρα:

Μοιράζεται το 2ο φύλλο εργασίας (είναι 2 σελίδες- το μοιράζουμε σε διαφορετικά φύλλα)

Δεν αποτελεί στόχο η υλοποίηση όλου του φύλλου εργασίας σε μια διδακτική ώρα.

Η δεύτερη σελίδα, μπορεί να δοθεί στους μαθητές, ως προαιρετική εργασία για το σπίτι

Σχολείο

Ημερομηνία.....

Όνοματεπώνυμο.....

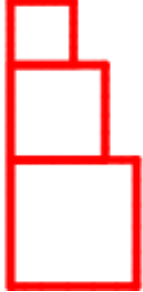
Τμήμα..... Αριθ. Καταλ.....

Συναρτήσεις με Μεταβλητές

☞ Έστω η παρακάτω συνάρτηση για τη δημιουργία τετραγώνου με μεταβλητό μήκος πλευράς. Αφού τη γράψετε στο περιβάλλον του Pencilcode, χρησιμοποιείστε τη για να δημιουργήσετε το παρακάτω σχήμα. Γράψτε στο φύλλο εργασίας τις εντολές που δώσατε για τη δημιουργία του. Τα τετράγωνα έχουν μήκη 40, 60 και 80 εικονοστοιχεία. Μπορείτε να το κάνετε με ότι χρώμα θέλετε.

```
tetragonol = (x) ->
  for [1..4]
    fd x
    rt 90
```

Χρησιμοποιήσατε πλακίδια ή κώδικα;.....

<p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p>	
--	---

☞ Δημιουργήστε μια συνάρτηση με το όνομα `kyklos1` για τη δημιουργία ενός κύκλου μαύρου χρώματος με μεταβλητή την ακτίνα του κύκλου. Σημειώστε τις εντολές στο φύλλο εργασίας, προσέχοντας τη σύνταξη.

Χρησιμοποιήσατε πλακίδια ή κώδικα;.....

.....

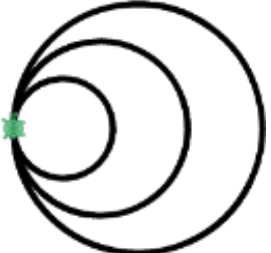
.....

.....

.....

☞ Χρησιμοποιείστε τη συνάρτηση `kyklos1` για τη δημιουργία του παρακάτω σχήματος. Οι κύκλοι έχουν ακτίνες 40, 60 και 80 εικονοστοιχεία αντίστοιχα. Σημειώστε τις εντολές στο φύλλο εργασίας.

Χρησιμοποιήσατε πλακίδια ή κώδικα;.....

	<p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p>
---	---

☞ Τροποποιήστε την παραπάνω συνάρτηση `kyklos3` ώστε να δέχεται μεταβλητές ΚΑΙ το χρώμα και το πάχος της γραμμής. Αποθηκεύστε τη νέα συνάρτηση με το όνομα `kyklos3`

.....
.....
.....
.....
.....
.....

☞ Να καλέσετε τη συνάρτηση `kyklos3` ώστε να εμφανιστούν:

i. ένας κόκκινος κύκλος με πάχος γραμμής 5 και ακτίνα 40 εικονοστοιχεία

.....

ii. ένας πράσινος κύκλος με πάχος γραμμής 10 και ακτίνα 60 εικονοστοιχεία

.....

iii. ένας μπλε κύκλος με πάχος γραμμής 15 και ακτίνα 80 εικονοστοιχεία

.....

Χρησιμοποιήσατε πλακίδια ή κώδικα;.....

Σχολείο

Ημερομηνία.....

Όνοματεπώνυμο.....

Τμήμα..... Αριθ. Καταλ.....

Συναρτήσεις με Μεταβλητές (2)

1. Να δημιουργήσετε μια συνάρτηση με το όνομα `isopleuro3` για το σχεδιασμό ισόπλευρου τριγώνου. (Μεταβλητές: το μήκος της πλευράς του τριγώνου, το πάχος της γραμμής και το χρώμα της γραμμής.)

.....
.....
.....
.....
.....
.....

Να καλέσετε τη συνάρτηση `isopleuro3` ώστε να εμφανιστούν:

i. ένα κόκκινο τρίγωνο με πλευρά 100 εικονοστοιχεία και πάχος 5

.....

ii. ένα πορτοκαλί τρίγωνο με πλευρά 200 εικονοστοιχεία και πάχος 4

.....

Χρησιμοποιήσατε πλακίδια ή κώδικα;.....

2. Γράψτε συνάρτηση με όνομα `polygono4` που θα παίρνει ως μεταβλητές: τον αριθμό των πλευρών του πολυγώνου, το μήκος, το χρώμα και το πάχος των πλευρών και θα σχηματίζει κανονικό πολύγωνο.

.....
.....
.....
.....
.....
.....

Πώς θα καλέσουμε τη συνάρτηση `polygono4` προκειμένου να σχεδιάσουμε τα παρακάτω;

i. ένα πράσινο τρίγωνο με πλευρά 100 εικονοστοιχεία και πάχος γραμμής 15

.....

ii. ένα κόκκινο τετράγωνο με πλευρά 120 εικονοστοιχεία και πάχος γραμμής 10

.....

iii. ένα μπλε εξάγωνο με πλευρά 80 εικονοστοιχεία και πάχος γραμμής 5

.....

Χρησιμοποιήσατε πλακίδια ή κώδικα;.....

3. Να δημιουργήσετε συνάρτηση με το όνομα `skala3` για τη δημιουργία μιας σκάλας που δέχεται ως μεταβλητές το πλήθος των σκαλοπατιών, το ύψος και το βάθος κάθε σκαλοπατιού, μαύρου χρώματος και πάχος γραμμής 5

.....

.....

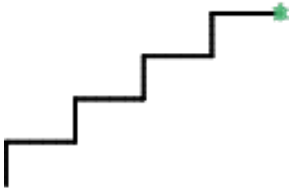
.....

.....

.....

.....

Πώς θα καλέσουμε τη συνάρτηση `skala3` προκειμένου να σχεδιάσουμε το παρακάτω σχήμα (ύψος σκαλιού 50 εικονοστοιχεία, βάθος 80 εικονοστοιχεία.

	<p>.....</p>
---	--------------

Χρησιμοποιήσατε πλακίδια ή κώδικα;.....

4. Να δημιουργήσετε μια συνάρτηση με το όνομα `ilios3` που θα σχεδιάζει ένα σχήμα όπως το παρακάτω. Η διαδικασία θα δέχεται σαν παραμέτρους τον αριθμό των ακτίνων και το μήκος της ακτίνας και το χρώμα της. Το πάχος της γραμμής να είναι 3.

.....

.....

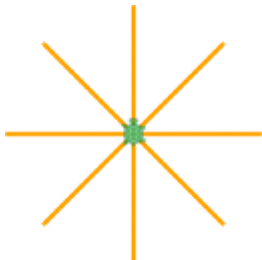
.....

.....

.....

.....

Πώς θα καλέσουμε τη συνάρτηση `ilios3` προκειμένου να σχεδιάσουμε το παρακάτω σχήμα (μήκος ακτίνας 100 εικονοστοιχεία, χρώμα πορτοκαλί.

	<p>.....</p>
---	--------------

Χρησιμοποιήσατε πλακίδια ή κώδικα;.....

13^η-14^η διδακτική ώρα : Η εντολή for με μεταβλητές από λίστα

Στόχοι

Στόχος του μαθήματος είναι οι μαθητές:

- Να δημιουργούν βρόχους επανάληψης με χρήση μεταβλητής που λαμβάνει τιμή από λίστα τιμών

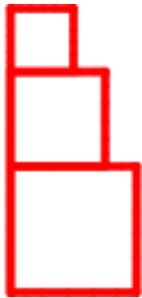
Σχέδιο μαθήματος

13^η ώρα:

Ο διδάσκων θυμίζει στους μαθητές μια προηγούμενη δραστηριότητα:

Σε προηγούμενη δραστηριότητα είχαμε δημιουργήσει συνάρτηση με το όνομα tetragono1 για τη δημιουργία τετραγώνου (η χελώνα αρχίζει και τελειώνει στην ίδια θέση) με μεταβλητή το μήκος της πλευράς.

Στη συνέχεια είχαμε καλέσει τη συνάρτηση για τη δημιουργία του παρακάτω σχήματος. Τα τετράγωνα έχουν μήκη 40, 60 και 80 εικονοστοιχεία.



Οι εντολές για να γίνει το παραπάνω σχήμα, καλώντας τη διαδικασία tetragono1 ήταν οι εξής:

```
tetragono1(80)
fd 80
tetragono1(60)
fd 60
tetragono1(40)
```

Ο διδάσκων με επίδειξη, δίνει στους μαθητές την εξής εναλλακτική (με κώδικα) ;

```
for x in [80, 60, 40]
    tetragono1(x)
    fd x
```

Το γράφει και στον πίνακα της τάξης.

Γίνεται μικρή συζήτηση για τη δυνατότητα της μεταβλητής να παίρνει τιμές από συγκεκριμένη λίστα τιμών.

Αναφέρεται η λέξη **Λίστα**, αλλά δεν εξηγούμε στην παρούσα φάση περισσότερα.

Μοιράζεται το 1^ο φύλλο εργασίας

14^η ώρα

Μοιράζεται το 2^ο φύλλο εργασίας

Σχολείο

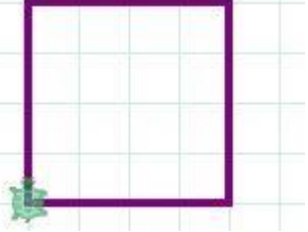
Ημερομηνία.....

Όνοματεπώνυμο.....

Τμήμα..... Αριθ. Καταλ.....

Μεταβλητές και η εντολή for (2)

- ☞ Δημιουργήστε μια συνάρτηση με το όνομα `tetragono1` για τη δημιουργία τετραγώνου (η χελώνα αρχίζει και τελειώνει στην ίδια θέση) με μεταβλητές το μήκος της πλευράς.

	<p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p>
---	--

- ☞ Χρησιμοποιείτε τη συνάρτηση `tetragono1` και την εντολή `for`, ώστε να δημιουργήσετε το επόμενο σχήμα: (τα τετράγωνα έχουν μήκη πλευρών 30, 40, 50, 60, 70, 80 εικονοστοιχεία)

Χρησιμοποιήσατε πλακίδια ή κώδικα;.....


	
--	--

- ☞ Δημιουργήστε μια συνάρτηση με το όνομα `kyklosc` για τη δημιουργία κύκλου ακτίνας 100 εικονοστοιχείων με μεταβλητή το χρώμα του

Χρησιμοποιήσατε πλακίδια ή κώδικα;.....

<p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p>
--

- ☞ Χρησιμοποιείτε τη συνάρτηση `kyklosc` και την εντολή `for`, ώστε να δημιουργήσετε το επόμενο σχήμα: (οι κύκλοι έχουν 6 διαφορετικά χρώματα)

	
---	--

- ☞ Προσπαθήστε να δημιουργήσετε ένα παραλληλόγραμμο με πλευρές 50 και 100 εικονοστοιχεία, χρησιμοποιώντας την εντολή `for` με μεταβλητή που θα παίρνει τιμές από μια λίστα.

Σχολείο

Ημερομηνία.....

Όνοματεπώνυμο.....

Τμήμα..... Αριθ. Καταλ.....

Μεταβλητές και η εντολή for

☞ Δοκιμάστε τις παρακάτω εντολές
`for x in [13, 35, 2]`
`write 'Κερδίζει ο αριθμός ', x`
Πόσες επαναλήψεις έγιναν; ____

☞ Δοκιμάστε τις παρακάτω εντολές
`for x in [0..4]`
`write 'ο αριθμός', x`
Πόσες επαναλήψεις έγιναν; ____
Ποια ήταν τιμή του x στην 1η επανάληψη; ____
Ποια ήταν τιμή του x στην τελευταία επανάληψη; ____

☞ Δοκιμάστε τις παρακάτω εντολές
`for x in [0..4]`
`write 'ο αριθμός', x`
Πόσες επαναλήψεις έγιναν; ____
Ποια ήταν τιμή του x στην 1η επανάληψη; ____
Ποια ήταν τιμή του x στην τελευταία επανάληψη; ____

☞ Δοκιμάστε τις παρακάτω εντολές
`for x in [10..20] by 2`
`write x`
Ποιοι αριθμοί εμφανίστηκαν; _____
Με ανάλογο τρόπο, ζητείστε να εμφανιστούν όλοι οι αριθμοί από το 1 έως το 100 που είναι πολλαπλάσια του 5. Γράψτε τις εντολές που χρησιμοποιήσατε:

☞ Χρησιμοποιείστε την εντολή for με μεταβλητή για να εμφανιστεί το παρακάτω. Γράψτε τις εντολές που χρησιμοποιήσατε

ο αριθμός 10 ο αριθμός 43 ο αριθμός 6 ο αριθμός 27	
---	--

☞ Χρησιμοποιείστε την εντολή for με μεταβλητή για να εμφανιστεί το παρακάτω. Γράψτε τις εντολές που χρησιμοποιήσατε.

Γεια σου Ελένη Γεια σου Γιάννη Γεια σου Μαρία	
---	--

☞ Δοκιμάστε τις παρακάτω εντολές

```
for color in [red, orange, yellow, blue]
  pen color, 4
  fd 100
  rt 90
```

☞ Δημιουργήστε με ανάλογο τρόπο ένα κανονικό οκτάγωνο με διαφορετικά χρώματα στην κάθε πλευρά.
Σημειώστε τις εντολές

☞ Δοκιμάστε τις παρακάτω εντολές

```
pen purple
for x in [50..1] by -1
  rt 30, x
```

☞ Τροποποιείστε τις παραπάνω εντολές ώστε η σπείρα να ξεκινήσει από μέσα προς τα έξω
Χρησιμοποιήστες Πλακίδια ή Κώδικα;.....

15^η διδακτική ώρα : Δημιουργώ μεταβλητές

ΣΤΟΧΟΙ

Στόχοι του μαθήματος είναι οι μαθητές:

- Να δημιουργούν μεταβλητές
- Να περιγράφουν την λειτουργία τους μέσα σε ένα πρόγραμμα
- Να προβλέπουν την τιμή μιας μεταβλητής μετ από την εκτέλεση του προγράμματος

Ο διδάσκων ζητά από τους μαθητές να κάνουν τις επόμενες δραστηριότητες.


Χρησιμοποιεί τον πίνακα Και γίνεται συζήτηση

Σχολείο

Ημερομηνία.....

Όνοματεπώνυμο..... Τμήμα..... Αριθ. Καταλ.....


Φύλλο εργασίας- δημιουργώ μεταβλητές

 Δοκιμάστε τις παρακάτω εντολές ΜΙΑΜΙΑ και γράψτε τι εμφανίζεται ή τι ακούγεται κάθε φορά. (αν δεν εμφανίζεται κάτι γράψτε -)


x= 3	
writex*x	
x='Maria'	
writex	
sayx	
write 'x'	

 Για να δώσω την τιμή 5 στην μεταβλητή ποσό ποια εντολή θα δώσω;

.....

 Για να εμφανίσω την τιμή της μεταβλητής ποια εντολή θα δώσω;


.....

 Σβήστε τις παραπάνω εντολές και δοκιμάστε πάλι την εντολή


writex

Τι παρατηρούμε;


Ακολουθεί συζήτηση


 Δοκιμάστε τις παρακάτω εντολές ΜΙΑΜΙΑ και γράψτε τι εμφανίζεται ή τι ακούγεται κάθε φορά. (αν δεν εμφανίζεται κάτι γράψτε -)

x=5	
x=x+3	
write x	
writex*2	
y=10	
for z in [1..5] write y*z	

 Δοκιμάστε τις παρακάτω εντολές ΜΙΑΜΙΑ και γράψτε τι εμφανίζεται ή τι ακούγεται κάθε φορά. (αν δεν εμφανίζεται κάτι γράψτε -)

x=7	
y=x+6	
write y	

 Δοκιμάστε τις παρακάτω εντολές
awaitread 'Πώς σε λένε;', deferx
alert 'Γεια σου ' + x

 Χρησιμοποιώντας την εντολή `await` , γράψτε μία διαδικασία με όνομα `Γινόμενο` η οποία όταν εκτελείται:

α) να ρωτάει με κατάλληλο μήνυμα ένα αριθμό και να τον αποθηκεύει σε μια μεταβλητή με το όνομα `A1`

β) να ρωτάει με κατάλληλο μήνυμα ένα δεύτερο αριθμό και να τον αποθηκεύει σε μια μεταβλητή με τον όνομα `A2`

γ) να υπολογίζει το γινόμενο των δύο αριθμών και να εκχωρεί τον αριθμό αυτό σε μία μεταβλητή με όνομα `Γ`.

δ) τέλος να βγάζει ένα παράθυρο όπου να εμφανίζεται μήνυμα :

το γινόμενο των αριθμών είναι `Γ`. (στη θέση `Γ` θα εμφανιστεί η τιμή του)

16^η-17^η ώρα: Δομή επιλογής

Στόχοι

Οι μαθητές να είναι ικανοί να

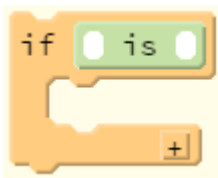
- Περιγράψουν τις διάφορες μορφές της δομής επιλογής
- Να επιλέγουν την κατάλληλη μορφή της δομής επιλογής στα προγράμματα που αναπτύσσουν

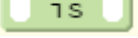
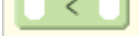
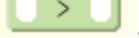
Εισαγωγή

Στον προγραμματισμό, όπως και στη ζωή, συχνά παρουσιάζεται το φαινόμενο να αλλάξουμε τις ενέργειες που πρέπει να κάνουμε ανάλογα αν ισχύει κάποια συνθήκη ή δεν ισχύει. Για παράδειγμα θέλουμε να επισκεφτούμε ένα φίλο μας αλλά δεν γνωρίζουμε αν είναι στο σπίτι του. Τότε η σκέψη μας είναι «Θα τηλεφωνήσω και αν τον βρω θα πάω να τον επισκεφτώ». Αν διαπιστώσουμε λοιπόν ότι βρίσκεται στο σπίτι του θα κάνουμε κάποιες συγκεκριμένες ενέργειες οι οποίες δεν θα γίνουν αν ο φίλος δεν βρεθεί. Έτσι και στον προγραμματισμό είναι συχνό το φαινόμενο να θέλουμε ένα τμήμα του προγράμματος με συγκεκριμένες εντολές να εκτελείται ή αν μην εκτελείται ανάλογα με το αν ισχύει κάποια συνθήκη.

Η διαδικασία αυτή στο PencilCode πραγματοποιείται με τη δομή επιλογής.

Η απλή δομή επιλογής έχει στα εικονίδια τη μορφή:



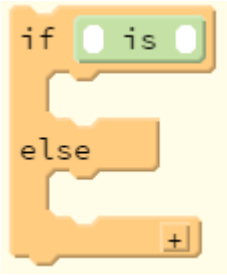
Η συνθήκη  μπορεί να αλλάξει και να γίνει  ή . Τα εικονίδια αυτά υφάρχουν στην ομάδα Operators.

Παράδειγμα: Αν ο αριθμός x είναι θετικός προχώρα μπροστά κατά x βήματα

Στα Εικονίδια	Στον Κώδικα
<pre>1 pen ▼ black 2 x = 100 3 if x > 0 4 fd ▼ x</pre>	<pre>pen black x = 100 if x > 0 fd x</pre>

Η παραπάνω δομή ονομάζεται απλή γιατί υπάρχουν ποιο σύνθετες μορφές της. Για παράδειγμα μπορούμε να καθορίσουμε από την αρχή τι θα γίνει αν δεν ισχύει η συνθήκη που βάλαμε. Στο παράδειγμα με το τηλεφώνημα στον φίλο μας η σκέψη μας θα μπορούσε να είναι «Θα τηλεφωνήσω στο φίλο μου και αν τον βρω θα πάω να τον επισκεφτώ αλλιώς (αν δεν τον βρω) θα παίξω στον υπολογιστή».

Η δομή αυτή έχει τη μορφή

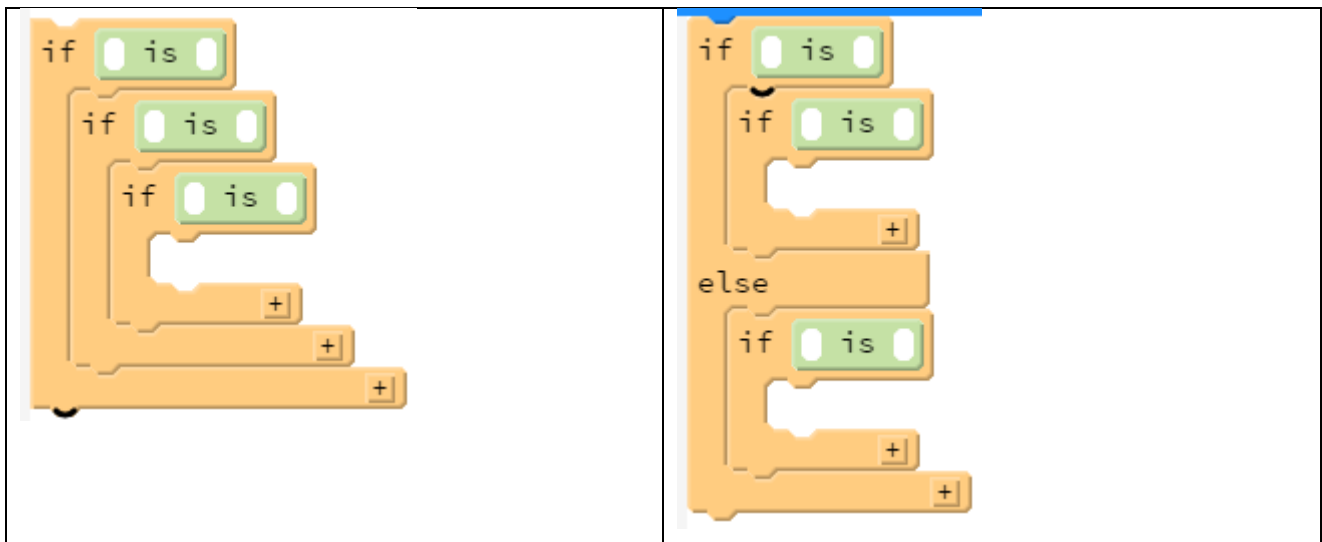


Παράδειγμα: Δώσε ένα αριθμό. Αν ο αριθμός x είναι θετικός προχώρα μπροστά κατά x βήματα αλλιώς (αν δεν είναι θετικός) πήγαινε προς τα πίσω κατά x βήματα

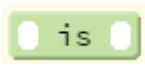
Στα Εικονίδια	Στον Κώδικα
	<pre> pen black await read 'Δώσε αριθμό βημάτων', defer x if x > 0 fd x else fd x </pre>

Παρατήρηση: Μετά το else παρατηρούμε ότι πάλι υπάρχει η εντολή fd (μπροστά) αλλά επειδή ο αριθμός είναι αρνητικός η χελώνα θα κινηθεί προς τα πίσω.

Η δομή επιλογής επιτρέπει την "εμφώλευση" δηλαδή την τοποθέτηση μιας δομής επιλογής μέσα σε μία άλλη. Για παράδειγμα



Αντίστοιχα σε μορφή κώδικα η εμφώλευση γίνεται με τις κατάλληλες εσοχές.

Η συνθήκη που βάζουμε μετά το If Μπορεί να είναι απλή όπως  π.χ x is 5, η σύνθετη. Μια σύνθετη συνθήκη υλοποιείται με τη βοήθεια λογικών συνδέσμων

Λογικοί σύνδεσμοι

Στο Pencil Code χρησιμοποιούνται οι λογικοί σύνδεσμοι **and**, **or** και **not** (Σε κώδικα αντίστοιχα `and`, `or` και `not`). Σε κάθε κενό ενός λογικού συνδέσμου μπορούμε να

τοποθετήσουμε μια απλή συνθήκη π.χ `x > 0 and y > 0`, `x is 1 or y is 1`. Σε κώδικα σύνθετες συνθήκες μπορεί να έχουν τη μορφή `x > 0 and y > 0, x is 1 or y is 1`.

Υπενθυμίζεται ότι η αλήθεια μιας λογικής συνθήκης που περιέχει ένα λογικό σύνδεσμο υπολογίζεται από τον παρακάτω πίνακα (όπου A και B συνθήκες π.χ `x > 0`).

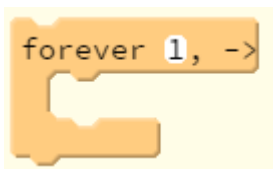
A	B	not A	not B	A and B	A or B
Αληθής	Αληθής	Ψευδής	Ψευδής	Αληθής	Αληθής
Αληθής	Ψευδής	Ψευδής	Αληθής	Ψευδής	Αληθής
Ψευδής	Αληθής	Αληθής	Ψευδής	Ψευδής	Αληθής
Ψευδής	Ψευδής	Αληθής	Αληθής	Ψευδής	Ψευδής

Ο πίνακας δίνεται για ενημέρωση του εκπαιδευτικού και δεν πρέπει αν διδαχθεί στους μαθητές. Στους μαθητές αρκεί να ειπωθεί ότι το `not` αντιστρέφει την συνθήκη, το `and` ισχύει όταν και οι δύο συνθήκες είναι αληθινές και το `or` ισχύει πάντα εκτός αν και οι δύο συνθήκες είναι ψευδείς. Μπορούν να χρησιμοποιηθούν παραδείγματα από την πραγματική ζωή π.χ Αύριο θα πάω κινηματογράφο και θα διαβάσω Ιστορία (αντίστοιχα θα πάω κινηματογράφο ή θα διαβάσω Ιστορία) πότε θα έχω πει ψέματα και πότε αλήθεια?.

Δομή επανάληψης forever

Η δομή επανάληψης `forever` χρησιμοποιείται όταν θέλουμε κάποια επανάληψη να εκτελείται συνεχώς. Για παράδειγμα μια δομή επανάληψης που ρυθμίζει ένα παιχνίδι. Η δομή `while` δεν είναι κατάλληλη για ατέρμονες επαναλήψεις γιατί η χρήση της θα προκαλέσει το "κρέμασμα" του φυλλομετρητή επειδή οι διαδοχικές επαναλήψεις γίνονται άμεσα η μία μετά την άλλη. Σε αντίθεση η `forever` κάνει μια μικρή διακοπή ανάμεσα στις επαναλήψεις και επιτρέπει την αλληλεπίδραση με το χρήστη. Ο αριθμός που βρίσκεται μετά το `forever` καθορίζει το κενό χρόνου που θα υπάρχει μεταξύ των επαναλήψεων. Συγκεκριμένα ο αριθμός εκφράζει πόσες επαναλήψεις θα εκτελεστούν σε ένα δευτερόλεπτο.

Η δομή `forever` βρίσκεται στην ομάδα πλακιδίων Control και είναι:



Διακοπή - τερματισμός της forever

Ένας ατέρμων βρόχος μπορεί να διακοπεί (και να τερματιστεί) με την εντολή `stop()`. Η εντολή αυτή θα πρέπει να τοποθετηθεί είτε σε μια δομή επιλογής `if`, είτε σε μια δομή ελέγχου του πληκτρολογίου όπως η `keydown`.

Παράδειγμα χρήσης

	<pre>forever 1, -> pen purple, 10 fd 30 rt 20 keydown 's', -> stop()</pre>
--	--

Στο παραπάνω παράδειγμα η εκτέλεση του προγράμματος σταματά μόλις πατηθεί το γράμμα "s" (Η επιλογή του γράμματος είναι τυχαία από το "stop")

Σχέδιο μαθήματος

1^η διδακτική ώρα

Ο Διδάσκων μοιράζει στους υπολογιστές των μαθητών ένα αρχείο με παραδείγματα αντιγράψτε το πρώτο Φύλλο εργασίας σε ένα αρχείο doc και μοιράστε το στους υπολογιστές, ώστε να είναι διαθέσιμο στους μαθητές, σε ηλ. Μορφή)

Το αρχείο περιέχει οδηγίες στους μαθητές για να δουλέψουν. Ενδεικτικές λύσεις:

```
button 'βουνο2', ->
  await read ' ποιο είναι το ψηλότερο βουνό της Ελλάδας', defer x
  if x is 'Όλυμπος'
    wear 'mountain'
  else
    wear 'wrong'
```

```
button 'βαθμολογία', ->
  await read 'βαθμός α τετραμήνου', defer t1
  await read 'βαθμός β τετραμήνου', defer t2
  await read 'βαθμός εξετάσεων', defer g
  V=(t1+t2+g)/3
  if V <10
    write 'πρέπει να ξαναδώσεις εξετάσεις'
    wear 'sad'
  else
    write 'πέρασες το μάθημα'
    wear 'congratulations'
```

2^η διδακτική ώρα

Μοιράζεται το φύλλο εργασίας. Ενδεικτικές λύσεις:

```
button 'πρωτοβάθμια', ->
  await read 'α', defer a
  await read 'β', defer b
  write 'η εξίσωση '+ a + ' * x = ' + b
  if a is 0
```

```
    if b is 0
        write ' η εξίσωση είναι αόριστη'
    else
        write ' η εξίσωση είναι αδύνατη'
else
    x= b/a
    write 'η εξίσωση έχει λύση ' + x

button 'πρωτοβαθμια2', ->
    await read 'α', defer a
    await read 'β', defer b
    if a is 0 and b is 0
        write 'η εξίσωση είναι αόριστη'
    else if a is 0 and b isnt 0
        write 'η εξίσωση είναι αδύνατη'
    else
        write 'η εξίσωση έχει λύση' + b / a
```

1^ο Φύλλο εργασίας- Η εντολή if

Αντιγράψτε τον παρακάτω κώδικα στο pencil code, κρατώντας την ίδια σύνταξη.

```
await read 'πόσο χρονών είσαι', defer x
if x > 17
  write 'είσαι ενήλικας'
else
  write 'είσαι παιδι'
```

Τρέξτε το πρόγραμμα 2-3 φορές δίνοντας ως απάντηση, διαφορετικές ηλικίες. Βάλτε τον προηγούμενο κώδικα σε ένα κουμπί:

```
button 'ηλικία', ->
  await read 'πόσο χρονων είσαι', defer x
  if x > 17
    write 'είσαι ενήλικας'
  else
    write 'είσαι παιδι'
```

Αντιγράψτε στο ίδιο αρχείο τον παρακάτω κώδικα:

```
button 'ρίξε το ζάρι', ->
  x = random [1..6]
  write 'το ζαρι έδειξε ' + x
  if x is 1
    write 'θα δουλέψω στο pencilcode.'
    wear 'pencil'
  else if x is 2
    write 'θα διαβάσω μαθηματικά!'
    wear 'math'
  else if x is 3
    write 'θα διαβάσω Λογοτεχνία.'
    wear 'book'
  else if x is 4
    write 'θα κάνω γυμναστική!'
    wear 'gym'
  else if x is 5
    write 'θα δω ταινία'
    wear 'cinema'
  else
    write 'θα πάω βόλτα'
    wear 'walk'
```

Πατήστε το κουμπί ρίξε το ζάρι

- ☞ Προσπαθήστε να φτιάξετε ένα κουμπί που θα εμφανίζει την ερώτηση : Ποιο είναι το ψηλότερο βουνό της Ελλάδας; Ο υπολογιστής θα ελέγχει την απάντηση που θα δώσει ο χρήστης. Αν απαντήσει σωστά, η χελώνα θα ντυθεί «mountain», αλλιώς θα ντυθεί «wrong».
- ☞ Προσπαθήστε να φτιάξετε να κουμπί που θα εμφανίζει ερωτήσεις ώστε ο χρήστης να δίνει τον βαθμό του στα μαθηματικά στο α τετράμηνο, στο β τετράμηνο και στις εξετάσεις. Αν ο μέσος όρος είναι μεγαλύτερος του 10, θα γράφεται το κείμενο «Πέρασες το μάθημα» Αλλιώς θα γράφεται το κείμενο «Πρέπει να ξαναδώσεις εξετάσεις».

Σχολείο

Ημερομηνία.....

Όνοματεπώνυμο..... Τμήμα..... Αριθ. Καταλ.....

2^ο Φύλλο εργασίας- η εντολή If



Έστω ο παρακάτω κώδικας στο Pencilcode. ΧΩΡΙΣ να τον αντιγράψετε στο περιβάλλον του pencilcode, συμπληρώστε στον παρακάτω πίνακα, τι μήνυμα θα εμφανιστεί σε κάθε μια από τις παρακάτω περιπτώσεις:

```
button 'μάθημα', ->
  await read ' πόσο είναι ο Μέσος όρος?', defer m
  await read ' πόσο είναι ο βαθμός στην Ιστορία?', defer h
  if m < 10 and h < 10
    write 'έμεινες στην ίδια τάξη'
  else if m > 10 and h > 10
    write 'πέρασες'
  else
    write 'ρώτα τον καθηγητή σου για το αποτέλεσμα'
```

Μέσος όρος	Βαθμός στην ιστορία	μήνυμα
16	17	
12	8	
9	12	
9	9	



Έστω ο παρακάτω κώδικας στο pencilcode. ΧΩΡΙΣ να τον αντιγράψετε στο περιβάλλον του pencilcode, συμπληρώστε στον παρακάτω πίνακα, ποιο θα είναι το κόστος μεταφοράς ενός οχήματος με το καράβι σε κάθε μια από τις παρακάτω περιπτώσεις:

```
button 'karavi', ->
  await read 'τι εποχή έχουμε?', defer s
  await read 'βάρος οχήματος', defer v
  if s is 'Καλοκαίρι'
    if v > 1000
      kostos = 100
    else
      kostos = 80
  else
    if v > 1000
      kostos = 70
    else
      kostos = 50
  write 'το κόστος μεταφοράς θα είναι ' + kostos + ' ευρώ'
```

εποχή	βάρος	κόστος
Καλοκαίρι	700	
Χειμώνας	1300	
Καλοκαίρι	1200	
Άνοιξη	600	



Προσπαθήστε να φτιάξετε ένα κουμπί που όταν το πατάμε, θα μας ζητάει να του δώσουμε τους συντελεστές της πρωτοβάθμιας εξίσωσης $ax=b$ και να μας δίνει την λύση της.

(θυμόμαστε από τα μαθηματικά: όταν η εξίσωση είναι της μορφής $0x=0$ η εξίσωση είναι ταυτότητα, όταν η εξίσωση έχει $a=0$ και $b \neq 0$ η εξίσωση είναι αδύνατη, σε κάθε άλλη περίπτωση η εξίσωση έχει λύση $x= b/a$)