

Ενότητα 2 Προγραμματισμός με την Γλώσσα Python

Στόχοι:

- Να κατανοήσουν οι μαθητές τη σύνταξη και τη λογική της Python.
 - Να αναπτύξουν αλγοριθμική σκέψη και δεξιότητες προγραμματισμού.
 - Να δημιουργήσουν απλές εφαρμογές και παιχνίδια.
-

Ωρα 1: Εισαγωγή στη Python και το EduBlocks

- Επισκόπηση της Python και εφαρμογών της.
 - Εγκατάσταση/πρόσβαση στο EduBlocks.
 - Σχεδιασμός απλών σχημάτων με τη βιβλιοθήκη turtle.
 - Δραστηριότητα: Σχεδιάστε ένα τετράγωνο και ένα τρίγωνο.
-

Ωρα 2: Τύποι Δεδομένων και Τελεστές

- Εξήγηση τύπων δεδομένων (int, float, str, bool).
 - Αριθμητικοί, συγκριτικοί και λογικοί τελεστές.
 - Δραστηριότητα: Υπολογισμοί στο shell της Thonny (π.χ. `2**10`, `"10"+"10"`).
-

Ωρα 3: Μεταβλητές και Εντολές Εισόδου/Εξόδου

- Δήλωση και χρήση μεταβλητών.
 - Εντολές `input` και `print`, μετατροπή τύπων (int, str).
 - Δραστηριότητα: Πρόγραμμα που ζητάει αριθμό και τον τυπώνει 10 φορές.
-

Ωρα 4: Συναρτήσεις και Υποπρογράμματα

- Ορισμός συναρτήσεων με `def`.
 - Παράμετροι και κλήση συναρτήσεων.
 - Δραστηριότητα: Συνάρτηση `Polygon(length, N)` για σχεδιασμό πολυγώνων.
-

Ωρα 5: Δομές Ελέγχου (if-else)

- Απλές και σύνθετες συνθήκες.
 - Χρήση των `if`, `elif`, `else`.
 - Δραστηριότητα: Πρόγραμμα ελέγχου άρτιου/περιττού αριθμού.
-

Ωρα 6: Επανάληψεις (for, while)

- Επανάληψη με `for` και `range`.
 - Επανάληψη με `while` και συνθήκη.
 - Δραστηριότητα: Εκτύπωση αριθμών 1 έως 10 με `for` και `while`.
-

Ωρα 7: Τυχειότητα και Παιχνίδια

- Βιβλιοθήκη `random`, `randint`.
 - Πρόγραμμα μαντεψίματος αριθμού ζαριάς.
 - Δραστηριότητα: Επέκταση με πολλαπλές προσπάθειες και `break`.
-

Ωρα 8: Μεγάλη Πρακτική Άσκηση

- Ανάπτυξη παιχνιδιού "Πέτρα-Ψαλίδι-Χαρτί".

- Χρήση λιστών, τυχαίας επιλογής, συνθηκών και επαναλήψεων.
- Παρουσίαση προγραμμάτων και αποτίμηση.

Αξιολόγηση

- Συμμετοχή στις δραστηριότητες.
 - Ολοκλήρωση απλών προγραμμάτων.
 - Τελικό πρόγραμμα (παιχνίδι).
-

Ενότητα 2: Προγραμματισμός με τη Γλώσσα Python

Σκοπός της Ενότητας: Να εισαγάγει τους μαθητές στις βασικές αρχές του προγραμματισμού με την Python, μια σύγχρονη και ισχυρή γλώσσα που χρησιμοποιείται σε πολλούς τομείς (εφαρμογές, AI, επιστήμη). Η έμφαση δίνεται στην κατανόηση εννοιών, στην επίλυση προβλημάτων και στη δημιουργία διαδραστικών προγραμμάτων.

Λεπτομερής Αναλυτική Περίληψη ανά Κεφάλαιο

1. Εισαγωγή & Η Γλώσσα Python (Σελ. 1-2)

- **Σκοπός:** Να δείξει τη σημασία και την ευρεία χρήση της Python στον πραγματικό κόσμο.
- **Βασικές Έννοιες:**
 - Η Python είναι γλώσσα υψηλού επιπέδου, εύκολη στη μάθηση.
 - Χρησιμοποιείται από μεγάλες εφαρμογές (Google, Netflix, Instagram), στην επιστήμη (αστροφυσική) και στην Τεχνητή Νοημοσύνη.
 - **Αντικειμενοστρεφής Προγραμματισμός:** Σε αντίθεση με το Scratch, δημιουργούμε αντικείμενα (π.χ. `franklin = Turtle()`) και τα «διατάζουμε» να εκτελέσουν εντολές (π.χ. `franklin.forward(100)`).
- **Δραστηριότητα (Σύγκριση):** Ομαδική συζήτηση για τις διαφορές μεταξύ ενός οπτικού προγραμματισμού (Scratch) και μιας κειμενικής γλώσσας (Python) βασισμένη στην εικόνα 2.3.

2. Το Περιβάλλον EduBlocks & Δεδομένα (Σελ. 3-7)

- **Σκοπός:** Να εξοικειώσει τους μαθητές με το κύριο περιβάλλον εργασίας (EduBlocks) και τις θεμελιώδεις έννοιες των δεδομένων.
- **Βασικές Έννοιες:**
 - **EduBlocks:** Οπτικό περιβάλλον (blocks) που παράγει κώδικα Python. Ιδανικό για αρχάριους. Εμφανίζει και την αντίστοιχη κειμενική σύνταξη.
 - **Τύποι Δεδομένων:**
 - `int` (ακέραιοι), `float` (πραγματικοί), `str` (αλφαριθμητικά/κείμενο), `bool` (Λογικές τιμές True/False).
 - `list` (λίστες): Ο σημαντικότερος σύνθετος τύπος. Δυναμική αποθήκευση πολλαπλών τιμών (π.χ. `["ΠΕΤΡΑ", "ΨΑΛΙΔΙ", "ΧΑΡΤΙ"]`).
 - **Τελεστές:**
 - **Αριθμητικοί:** `+`, `-`, `*`, `/`, `//`, `%`, `**` (με ιεραρχία πράξεων).

- **Συγκριτικοί:** `==`, `!=`, `<`, `>`, `<=`, `>=` (επιστρέφουν `True/False`).
- **Λογικοί:** `and`, `or`, `not` (για σύνθετες συνθήκες).
- **Μετατροπή Τύπων:** Κρίσιμη έννοια κατά την είσοδο δεδομένων. Η `input()` πάντα επιστρέφει κείμενο (`str`). Χρησιμοποιούμε `int()`, `float()`, `str()` για μετατροπή.
- **Πρακτικό Μάθημα:** Εκτέλεση των δραστηριοτήτων 1,2,3 στο shell της Thonny για να δουν οι μαθητές άμεσα τα αποτελέσματα των πράξεων, των τελεστών και των συναρτήσεων `len()` και `str()`.

3. Μεταβλητές & Είσοδος/Εξοδος (Σελ. 8-9)

- **Σκοπός:** Κατανόηση του ρόλου των μεταβλητών ως «περιεκτικών» για δεδομένα και της αλληλεπίδρασης με τον χρήστη.
- **Βασικές Έννοιες:**
 - **Μεταβλητές:** Συμβολικά ονόματα που «δείχνουν» σε δεδομένα στη μνήμη. Π.χ. `length = 10`.
 - **Κανόνες Ονομασίας:** Δεν ξεκινούν με αριθμό, δεν είναι δεσμευμένες λέξεις, περιγραφικά ονόματα.
 - **Τελεστής Ανάθεσης:** `=` («να γίνει»). Π.χ. `length += 3` (αύξηση κατά 3).
 - **Εντολές:** `print()` για έξοδο, `input()` για είσοδο.
- **Παράδειγμα:** Ανάλυση του προγράμματος της σπείρας (Εικόνα 2.5) για να κατανοήσουν πώς μια μεταβλητή (`length`) αλλάζει τιμή μέσα σε μια επανάληψη, αλλάζοντας και την έξοδο του προγράμματος.

4. Υποπρογράμματα - Συναρτήσεις (Σελ. 10)

- **Σκοπός:** Εισαγωγή στην αρχή της «δημιουργίας δικών μας εντολών» για να οργανώσουμε και να επαναχρησιμοποιήσουμε κώδικα.
- **Βασικές Έννοιες:**
 - **Ορισμός Συνάρτησης:** `def όνομα_συνάρτησης(παράμετροι):`
 - **Παράμετροι:** Μεταβλητές που δίνουμε στη συνάρτηση για να δουλέψει (π.χ. `length`, `N sides`).
 - **Κλήση Συνάρτησης:** `όνομα_συνάρτησης(τιμές)` για να την εκτελέσουμε.
 - **Χρήση της turtle:** Σχεδίαση πολυγώνων με βάση παραμέτρους.
- **Δραστηριότητα:** Οι μαθητές ορίζουν τη συνάρτηση `Polygon()` και στη συνέχεια καλούν την `Polygon(100, 6)` για να σχεδιάσουν ένα εξάγωνο. Προσπαθούν να λύσουν τις προτεινόμενες δραστηριότητες για ένθετα πολύγωνα.

5. Δομές Ελέγχου & Τυχαιότητα (Σελ. 11-15)

- **Σκοπός:** Να μάθουν οι μαθητές να ελέγχουν τη ροή του προγράμματος (συνθήκες, επαναλήψεις) και να δημιουργούν μη ντετερμινιστικά προγράμματα (παιχνίδια).
- **Βασικές Έννοιες:**
 - **Τυχαιότητα:** `import random`, `random.randint(1, 6)` για παραγωγή τυχαίων αριθμών.
 - **Συνθήκες (if-elif-else):** Εκτέλεση διαφορετικών εντολών ανάλογα με συνθήκες (`True/False`).
 - **Επανάληψεις:**
 - `for i in range(4):` Όταν γνωρίζουμε το πλήθος των επαναλήψεων.
 - `while condition:` Όταν η επανάληψη εξαρτάται από μια συνθήκη.
 - `break:` Για πρόωρο τέλος επανάληψης.

- ο **Λίστες & Δείκτης**: Πρόσβαση σε στοιχεία λίστας με δείκτη (index). Π.χ. `RPS[0]` είναι το "ΠΕΤΡΑ".
 - **Μεγάλο Έργο**: Ανάπτυξη του παιχνιδιού «Πέτρα-Ψαλίδι-Χαρτί» (Rock-Paper-Scissors). Συνδυάζει **όλες τις έννοιες**: λίστες, τυχαιότητα (randint), είσοδο (input), συνθήκες (if), επαναλήψεις (for/while), και μεταβλητές για το σκορ.
-

Συνολική Αξιολόγηση της Ενότητας

- **Δυνατά Σημεία**: Η ενότητα είναι **εξαιρετικά πρακτική και κυκλική**. Οι μαθητές ξεκινούν με απλά πράγματα (τετράγωνο με turtle) και φτάνουν να φτιάξουν ένα πλήρες, διαδραστικό παιχνίδι, βιώνοντας την πρόοδό τους. Η χρήση του EduBlocks είναι **ιδανική γέφυρα** από τον οπτικό προγραμματισμό (Scratch) στον κειμενικό (Python).
 - **Προκλήσεις**: Η μετάβαση από τα blocks στην καθαρή Python μπορεί να είναι απότομη για κάποιους μαθητές. Απαιτείται προσοχή στην **σωστή στοίχιση (indentation)** και στη **σύνταξη**. Οι έννοιες των συναρτήσεων και των παραμέτρων μπορεί να χρειαστούν επιπλέον χρόνο για να αφομοιωθούν.
 - **Σύσταση**: Να δοθεί έμφαση στη **διαδικαστική σκέψη** και στην **αναπαράσταση αλγορίθμων** (π.χ., με flowcharts) πριν την κωδικοποίηση, especialmente για τα πιο σύνθετα προγράμματα όπως το παιχνίδι.
-

Εννοιολογικός Χάρτης - Ενότητα 2: Python

Κεντρική Ιδέα: Η Python είναι μια ισχυρή και εύκολη γλώσσα προγραμματισμού για την ανάπτυξη εφαρμογών και την επίλυση προβλημάτων.

1. Εισαγωγή & Η Γλώσσα Python

- **Κύρια Ιδέα:** Γνωριμία με τη γλώσσα και το πλαίσιο εφαρμογών της.
 - **Έννοιες:**
 - **Python:** Γλώσσα Υψηλού Επιπέδου, Ευρεία Χρήση (Εφαρμογές, ΑΙ, Επιστήμη).
 - **Αντικειμενοστρεφής Προγραμματισμός (OOP):**
 - **Αντικείμενο (Object):** `franklin = Turtle()`
 - **Μέθοδος (Method):** `franklin.forward(100)`, `franklin.left(90)`
 - **Βιβλιοθήκες:** `from turtle import *`
 - **Σχέσεις:** Η Python $\neq$ Scratch. Χρησιμοποιεί αντικείμενα και μεθόδους αντί για απλά μπλοκ.
-

2. Το Περιβάλλον EduBlocks & Δεδομένα

- **Κύρια Ιδέα:** Το περιβάλλον ανάπτυξης και τα θεμελιώδη δομικά στοιχεία των δεδομένων.
 - **Έννοιες:**
 - **Περιβάλλον Ανάπτυξης (IDE):** EduBlocks (Blocks $\rightarrow$ Python), Thonny, IDLE.
 - **Τύποι Δεδομένων:**
 - **Βασικοί:** `int` (Ακέραιος), `float` (Πραγματικός), `str` (Συμβολοσειρά), `bool` (Λογικός).
 - **Σύνθετος:** `list` (Λίστα) $\rightarrow$ `["ΠΙΕΤΡΑ", "ΨΑΛΙΔΙ", "ΧΑΡΤΙ"]`
 - **Τελεστές (Operators):**
 - **Αριθμητικοί:** `+`, `-`, `*`, `/`, `//`, `%`, `**` (Ιεραρχία Πράξεων)
 - **Σύγκρισης:** `==`, `!=`, `<`, `>`, `<=`, `>=` $\rightarrow$ Επιστρέφουν `bool`
 - **Λογικοί:** `and`, `or`, `not` $\rightarrow$ Συνδυασμός Συνθηκών
 - **Μετατροπή Τύπου:** `int()`, `str()`, `float()` $\rightarrow$ Κρίσιμη για την `input()`
 - **Σχέσεις:** Το EduBlocks μας βοηθά να μάθουμε. Οι Τελεστές δρουν πάνω σε Δεδομένα συγκεκριμένου Τύπου.
-

3. Μεταβλητές & Είσοδος/Εξοδος

- **Κύρια Ιδέα:** Αποθήκευση δεδομένων και αλληλεπίδραση με τον χρήστη.
- **Έννοιες:**
 - **Μεταβλητή:** Συμβολικό όνομα για μια τιμή στη μνήμη (π.χ., `length = 10`).
 - **Κανόνες Ονομασίας:** Όχι αριθμός στην αρχή, όχι δεσμευμένες λέξεις, περιγραφική.
 - **Τελεστής Ανάθεσης:** `=` ("να γίνει"), `+=`, `-=`, `*=`, `/=`.
 - **Είσοδος:** `input("Μήνυμα:")` $\rightarrow$ Πάντα επιστρέφει `str`.
 - **Έξοδος:** `print("Κείμενο", variable)`.
- **Σχέσεις:** Η `input()` χρειάζεται συχνά **Μετατροπή Τύπου**. Οι **Μεταβλητές** κρατούν τα δεδομένα για επεξεργασία και **Έξοδο**.

4. Υποπρογράμματα - Συναρτήσεις

- **Κύρια Ιδέα:** Δημιουργία επαναχρησιμοποιήσιμων "μπλοκ" κώδικα για καλύτερη οργάνωση.
- **Έννοιες:**
 - **Συνάρτηση (Function):** `def function_name(parameters):`
 - **Παράμετροι (Parameters):** Μεταβλητές εισόδου της συνάρτησης (π.χ., `length, N`).
 - **Κλήση (Call):** `function_name(arguments) -> Polygon(100, 6)`
 - **Επανάληψη σε Συναρτήσεις:** `for i in range(N):` μέσα στη συνάρτηση.
- **Σχέσεις:** Οι **Συναρτήσεις** χρησιμοποιούν **Μεταβλητές** και **Παραμέτρους**. Μπορούν να καλούνται από το κύριο πρόγραμμα για **Έξοδο** (π.χ., σχεδίαση σχήματος).

5. Δομές Ελέγχου & Τυχαιότητα

- **Κύρια Ιδέα:** Ελέγχουμε τη ροή εκτέλεσης του προγράμματος και δημιουργούμε μη ντετερμινιστικές εφαρμογές.
- **Έννοιες:**
 - **Τυχαιότητα:** `import random, random.randint(min, max)`
 - **Δομή Επιλογής:**
 - `if / elif / else:` -> Εκτέλεση κώδικα βάσει **Συνθήκης** (Boolean έκφραση).
 - **Δομή Επανάληψης:**
 - `for i in range(start, stop, step):` -> Για γνωστό πλήθος επαναλήψεων.
 - `while condition:` -> Όσο η συνθήκη είναι True.
 - `break` -> Πρόωρος τερματισμός επανάληψης.
 - **Λίστες & Δείκτης:** `list_name[index]` για πρόσβαση σε στοιχεία.
- **Σχέσεις:** Η **Τυχαιότητα** τροφοδοτεί τις **Συνθήκες**. Οι **Επαναλήψεις** (`for/while`) ελέγχονται από **Συνθήκες** ή μετρητές. Όλες οι έννοιες συνδυάζονται για τη δημιουργία **Παιχνιδιών** (Rock-Paper-Scissors).

Συνολική Σχέση όλων των Εννοιών:

Ο πυρήνας του προγραμματισμού στην Python για αυτή την ενότητα είναι:

1. **Δεδομένα** (με συγκεκριμένο **Τύπο**) αποθηκεύονται σε **Μεταβλητές**.
2. Οι **Μεταβλητές** χειρίζονται με **Τελεστές** και **Συναρτήσεις**.
3. Η **ροή** του προγράμματος ελέγχεται με **Συνθήκες** (`if`) και **Επαναλήψεις** (`for/while`).
4. Η **αλληλεπίδραση** γίνεται με **Είσοδο** (`input`) και **Έξοδο** (`print`).
5. Ο **κώδικας** οργανώνεται σε **Συναρτήσεις** για να είναι πιο κατανοητός και επαναχρησιμοποιήσιμος.
6. Η **Τυχαιότητα** (`random`) κάνει τα προγράμματα απρόβλεπτα και διαδραστικά.

Αυτός ο εννοιολογικός χάρτης μπορεί να χρησιμοποιηθεί ως αφηρημένο "roadmap" από τον εκπαιδευτικό και ως φύλλο εργασίας/αναφοράς για τους μαθητές.

ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 1/5: Εισαγωγή & Η Γλώσσα Python

Σκοπός: Να γνωρίσουμε τη γλώσσα Python και τη βασική της σύνταξη.

Θεωρία - Σύντομη Επανάληψη:

- Η **Python** είναι μια δημοφιλής γλώσσα προγραμματισμού υψηλού επιπέδου.
- Χρησιμοποιείται για: Εφαρμογές, Ιστοσελίδες, Επιστήμη, Τεχνητή Νοημοσύνη.
- Σε αντίθεση με το Scratch, η Python είναι **κειμενική** και **αντικειμενοστρεφής**. Αυτό σημαίνει ότι δημιουργούμε αντικείμενα (π.χ. μια χελώνα) και της λέμε τι να κάνει.

1. Μετάφραση Κώδικα:

Μεταφράστε τον κώδικα Scratch στον αντίστοιχο κώδικα Python.

Scratch	Python
όταν γίνει κλικ σε	# Το πρόγραμμα ξεκινάει εδώ
επανάλαβε 4	for i in range(4):
κινήσου 100 βήματα	turtle.forward(100)
στρίψε 90 μοίρες	turtle.left(90)

2. Ποιος είναι ο Προγραμματιστής;

Συμπληρώστε τα κενά για να δημιουργήσετε ένα αντικείμενο χελώνας και να της ζητήσετε να σχεδιάσει ένα τετράγωνο.

python

from turtle import *

sophia = _____() # Δημιούργησε μια χελώνα ονόματι sophia

sophia._____() # Ζήτησέ της να κατεβάσει το στυλό της

for i in range(4):

 sophia._____ (100) # Προχώρα μπροστά 100 βήματα

 sophia._____ (90) # Στρίψε αριστερά 90 μοίρες

Πρακτική Δραστηριότητα:

Ανοίξτε το Thonny ή το EduBlocks. Γράψτε τον κώδικα από την άσκηση 2 και εκτελέστε τον. Τι σχήμα σχεδιάζει; Αλλάξτε το 90 σε 120. Τι σχήμα σχεδιάζει τώρα;

ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 2/5: Το Περιβάλλον EduBlocks & Δεδομένα

Σκοπός: Να εξοικειωθούμε με τους τύπους δεδομένων και τους τελεστές.

Θεωρία - Σύντομη Επανάληψη:

- **Τύποι Δεδομένων:** `int` (ακέραιος), `float` (δεκαδικός), `str` (κείμενο), `bool` (True/False).
- **Τελεστές:**
 - `+`, `-`, `*`, `/`, `**` (ύψωση σε δύναμη)
 - `==` (ίσο), `!=` (διάφορο), `>` (μεγαλύτερο), `<` (μικρότερο)
 - `and`, `or`, `not`

1. Ποιος είναι ο Τύπος;

Για κάθε τιμή, γράψτε τον τύπο της (`int`, `float`, `str`, `bool`).

α) 42, β) "Γεια σου!", γ) 3.14, δ) True, ε) "100"

2. Υπολογίστε τις Εκφράσεις:

Τι θα εκτυπώσει η Python;

```
python
print(10 + 5 * 2)    # Απάντηση: _____
print((10 + 5) * 2)  # Απάντηση: _____
print(2 ** 4)        # Απάντηση: _____
print(10 % 3)        # Απάντηση: _____ (Υπόλοιπο ακέραιας διαίρεσης)
print("10" + "5")    # Απάντηση: _____
print(10 + int("5")) # Απάντηση: _____
```

3. Αληθής ή Ψευδής;

Τι τιμή (True ή False) επιστρέφουν αυτές οι εκφράσεις;

```
python
print(10 > 5 and 3 < 1) # Απάντηση: _____
print(10 > 5 or 3 < 1)  # Απάντηση: _____
print(not(5 == 5))      # Απάντηση: _____
```

Πρακτική Δραστηριότητα:

Ανοίξτε το shell του Thonny. Δοκιμάστε μόνοι σας τις εκφράσεις από τις ασκήσεις 2 και 3 για να επαληθεύσετε τις απαντήσεις σας.

ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 3/5: Μεταβλητές & Είσοδος/Εξοδος

Σκοπός: Να μάθουμε να αποθηκεύουμε δεδομένα σε μεταβλητές και να αλληλεπιδρούμε με το χρήστη.

Θεωρία - Σύντομη Επανάληψη:

- **Μεταβλητή:** Είναι ένα "κουτί" που φυλάγει μια τιμή. Π.χ. `score = 0`
- `=` είναι ο τελεστής **ανάθεσης** (δεν σημαίνει "ίσο" όπως στα μαθηματικά).
- `input()`: Ζητάει είσοδο από το χρήστη. **Πάντα επιστρέφει κείμενο (string)!**
- `print()`: Εμφανίζει μηνύματα στην οθόνη.

1. Βρείτε το Λάθος:

Ποια από αυτές τις γραμμές κώδικα έχουν λάθος στη δήλωση της μεταβλητής; Γιατί;

```
python
1st_player = "Maria" # Λάθος: _____
score = 100          # Σωστό
player name = "Nikos" # Λάθος: _____
city = "Athens"      # Σωστό
```

2. Τι θα Εκτυπωθεί;

Τι θα εμφανίσει η οθόνη όταν τρέξει αυτό το πρόγραμμα;

```
python
name = input("Πώς σε λένε; ")
age = input("Πόσων ετών είσαι; ")
print("Γεια σου,", name, "! Είσαι", age, "ετών.")
# Αν ο χρήστης γράψει "Anna" και "14", η οθόνη θα δείξει:
```

3. Μετατροπή Τύπων:

Διορθώστε το πρόγραμμα για να λειτουργεί σωστά.

```
python
```



```
# Πρόγραμμα που ζητάει το έτος γέννησης και υπολογίζει την ηλικία
birth_year = input("Ποιο έτος γεννήθηκες; ")
age = 2024 - birth_year # ΕΔΩ ΕΙΝΑΙ ΤΟ ΛΑΘΟΣ!
print("Είσαι", age, "ετών.")
# ΔΙΟΡΘΩΣΗ: age = 2024 - _____ (birth_year)
```

Πρακτική Δραστηριότητα:

Γράψτε ένα πρόγραμμα που ζητάει από το χρήστη δύο αριθμούς και τυπώνει το άθροισμά τους.

ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 4/5: Υποπρογράμματα - Συναρτήσεις

Σκοπός: Να μάθουμε να οργανώνουμε τον κώδικα σε συναρτήσεις.

Θεωρία - Σύντομη Επανάληψη:

- **Συνάρτηση:** Είναι ένα μπλοκ κώδικα που εκτελεί μια συγκεκριμένη εργασία.
- **Ορισμός Συνάρτησης:** Ξεκινάμε με τη λέξη `def` και δύο τελείες (:).
- **Παράμετροι:** Είναι οι μεταβλητές που περνάμε στη συνάρτηση για να δουλέψει.
- **Κλήση Συνάρτησης:** Γράφουμε το όνομά της με τις απαιτούμενες τιμές (ορίσματα).

1. Δημιουργία Συνάρτησης:

Δημιουργήστε μια συνάρτηση ονόματι `τετράγωνο()` που να δέχεται μία παράμετρο `πλευρά` και να σχεδιάζει ένα τετράγωνο με αυτό το μήκος πλευράς.

```
python
from turtle import *
t = Turtle()

def tetragono(plevra):
    for i in range(4):
        t.forward(_____) # Συμπλήρωσε την εντολή
        t.left(_____)    # Συμπλήρωσε την εντολή
```

```
# ΚΑΛΕΣΕ ΤΗ ΣΥΝΑΡΤΗΣΗ ΓΙΑ ΝΑ ΣΧΕΔΙΑΣΕΙ ΤΕΤΡΑΓΩΝΟ ΠΛΕΥΡΑΣ 100
_____(100)
```

2. Πολυγωνική Τέχνη:

Ποιο σχήμα θα σχεδιάσει αυτό το πρόγραμμα; _____

```
python
from turtle import *
def polygon(sides, length):
    for i in range(sides):
        forward(length)
        left(360 / sides)
```

```
polygon(6, 100) # Σχεδιάζει ένα: _____
```

Πρακτική Δραστηριότητα:

Στο EduBlocks ή στο Thonny, γράψτε μια συνάρτηση `τρίγωνο(πλευρά)` που σχεδιάζει ισόπλευρο τρίγωνο. Χρησιμοποιήστε τον τύπο `left(360 / 3)` ή `left(120)`.

ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 5/5: Δομές Ελέγχου & Τυχειότητα

Σκοπός: Να ελέγξουμε τη ροή του προγράμματος με βάση συνθήκες και να χρησιμοποιήσουμε την τυχειότητα.

Θεωρία - Σύντομη Επανάληψη:

- **Συνθήκες (if/else):** Εκτελούν διαφορετικό κώδικα ανάλογα με το αν μια συνθήκη είναι αληθής (True) ή ψευδής (False).
- **Επανάληψη (for/while):** Εκτελούν έναν κώδικα πολλές φορές.
- **Τυχειότητα (random):** `import random` και `random.randint(1, 6)` για τυχαίο αριθμό.

1. Βήμα-Βήμα Ανάλυση:

Διαβάζοντας τον κώδικα, τι θα κάνει ο παίκτης στην τρίτη προσπάθεια αν ο υπολογιστής έχει επιλέξει το 4;

python

```
import random
```

```
secret = random.randint(1, 6)
```

```
found = False
```

```
attempts = 0
```

```
while not found and attempts < 3:
```

```
    guess = int(input("Μάντεψε (1-6): "))
```

```
    attempts += 1
```

```
    if guess == secret:
```

```
        print("Σωστά!")
```

```
        found = True
```

```
    else:
```

```
        print("Λάθος!")
```

```
if not found:
```

```
    print("Έχασες! Ο αριθμός ήταν", secret)
```

2. Rock-Paper-Scissors:

Συμπληρώστε τον κώδικα για το κλασικό παιχνίδι.

python

```
import random
```

```
choices = ["ΠΕΤΡΑ", "ΨΑΛΙΔΙ", "ΧΑΡΤΙ"]
```

```
computer_choice = choices[_____] # Γράψτε κώδικα για τυχαία επιλογή
```

```
player_choice = input("Διάλεξε (ΠΕΤΡΑ, ΨΑΛΙΔΙ, ΧΑΡΤΙ): ")
```

```
if player_choice == computer_choice:
```

```
    print("Ισόπαλο!")
```

```
elif (player_choice == "ΠΕΤΡΑ" and computer_choice == "ΨΑΛΙΔΙ") or (player_choice == "ΨΑΛΙΔΙ" and computer_choice == "ΧΑΡΤΙ") or (player_choice == "ΧΑΡΤΙ" and computer_choice == "ΠΕΤΡΑ"):
```

```
    print("Κέρδισες! Ο υπολογιστής διάλεξε", computer_choice)
```

```
else:
```

```
print("Έχασες! Ο υπολογιστής διάλεξε", computer_choice)
```

Πρακτική Δραστηριότητα:

Γράψτε ένα πρόγραμμα που "ρίχνει" ένα ζάρι (τυχαίος αριθμός 1-6) και ζητάει από το χρήστη να το μαντέψει. Χρησιμοποιήστε μια δομή if/else για να πείτε στον χρήστη αν βρήκε σωστά ή όχι.
