Περιβάλλοντα και γλώσσες προγραμματισμού

1. Προγραμματιστικά Περιβάλλοντα

2. Μικρό Εγχειρίδιο της γλώσσας προγραμματισμού Python

3. Μικρό Εγχειρίδιο της γλώσσας προγραμματισμού Javascript

1. Προγραμματιστικά Περιβάλλοντα

Ένα *Προγραμματιστικό Περιβάλλον* (ΠΠ) είναι ένα λογισμικό που παρέχει στους προγραμματιστές τα απαραίτητα εργαλεία για την ανάπτυξη εφαρμογών.

Είδη Προγραμματιστικών Περιβαλλόντων

Ολοκληρωμένα Περιβάλλοντα Ανάπτυξης (IDEs): Παρέχουν όλα τα απαραίτητα εργαλεία σε ένα ενιαίο περιβάλλον. Π.χ. Visual Studio, Eclipse, IntelliJ IDEA.

Επεξεργαστές κειμένου με δυνατότητες προγραμματισμού: Προσφέρουν βασικές λειτουργίες επεξεργασίας κώδικα και σύνδεσης με compiler/interpreter. Π.χ. Sublime Text, Atom, Notepad++.

Διαδικτυακά περιβάλλοντα ανάπτυξης: Παρέχουν δυνατότητα ανάπτυξης και εκτέλεσης προγραμμάτων μέσω ενός web browser. Π.χ. CodePen, JSFiddle, Glitch.

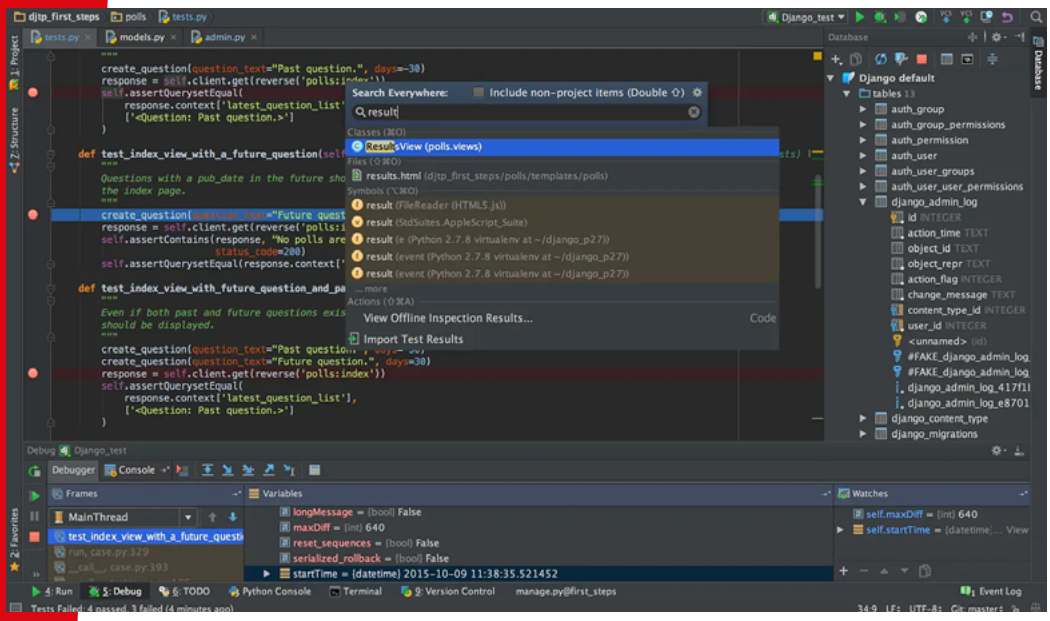
Τα σύγχρονα ολοκληρωμένα προγραμματιστικά περιβάλλοντα περιλαμβάνουν όλα τα κύρια εργαλεία που χρειάζεται ένας προγραμματιστής για να δημιουργήσει τις εφαρμογές του, όπως:

- **Συντάκτη** (editor) για την συγγραφή του προγράμματος. Ο συντάκτης μπορεί να είναι απλός συντάκτης ή να συνοδεύεται από επιπρόσθετη βοήθεια όπως προτάσεις για αυτόματη σύνταξη εντολών,, χρωματισμός κώδικα, αυτόματη συμπλήρωση εσοχών κλπ.
- **Εργαλεία Μετάφρασης και Ελέγχου:** Το εργαλείο μετάφρασης–ελέγχου μπορεί να είναι ένας μεταγλωττιστής (compiler) ή ένας διερμηνευτής (Interpreter). Σε σύγχρονα ΠΠ το εργαλείο μετάφρασης και ελέγχου λειτουργεί και ως διερμηνευτής αλλά και ως μεταγλωττιστής για την παραγωγή εκτελέσιμου κώδικα. Στο πρώτο στάδιο το εργαλείο μετάφρασης και ελέγχου εξετάζει τον κώδικα του προγράμματος, εντοπίζει συντακτικά λάθη και συντακτικές ελλείψεις και κάνει προτάσεις σχετικά με την διόρθωση των λαθών. Εφόσον το πρόγραμμα δεν έχει συντακτικά λάθη και παραλείψεις ακολουθεί η σύνδεση του προγράμματος με εξωτερικές βιβλιοθήκες και άλλες εργασίες μέχρι την μετατροπή του προγράμματος σε γλώσσα μηχανής, δηλαδή σε μια μορφή που μπορεί να εκτελεστεί απευθείας από τον υπολογιστή.
- **Βοηθητικά εργαλεία:** Τέτοια εργαλεία είναι η τεκμηρίωση με εμφάνιση βοήθειας και παραδειγμάτων όπως επίσης και βοηθούς προγραμματισμού που στηρίζονται στην Τεχνητή Νοημοσύνη.

Τα Προγραμματιστικά Περιβάλλοντα μπορούν να είναι εμπορικά λογισμικά ή ελεύθερα. Μπορούν να απευθύνονται σε μια συγκεκριμένη γλώσσα προγραμματισμού ή σε ένα σύνολο από γλώσσες προγραμματισμού. Ορισμένα από αυτά διαθέτουν γραφικά περιβάλλοντα και επίσης έχουν τη δυνατότητα να συνεργάζονται με άλλες εφαρμογές.

Στα Προγραμματιστικά Περιβάλλοντα ανήκουν και οι προγραμματιστικοί μικρόκοσμοι που έχουν αναπτυχθεί κυρίως για εκπαιδευτικούς σκοπούς όπως τα: Logo, Scratch, Snap, Kodu, Pencil code, Game maker, Alice, App Inventor κλπ.

Γνωστά περιβάλλοντα προγραμματισμού είναι τα Visual Studio, Eclipse, IntelliJ IDEA, NetBeans, C++Builder, PyCharm κλπ.



Εικόνα 1: Το Περιβάλλον προγραμματισμού PyCharm (source: https://www.jetbrains.com/pycharm/img/screenshots/complexLook@2x.jpg)

2. Μικρό Εγχειρίδιο της γλώσσας προγραμματισμού Python

Η Python είναι μια πολύ δημοφιλής γλώσσα προγραμματισμού τόσο για αρχάριους όσο και έμπειρους προγραμματιστές λόγω της απλότητάς της και της ευελιξίας της.

Εγκατάσταση Python:

1. Κατέβασε την Python από την επίσημη ιστοσελίδα: https://www.python.org/.
2. Ακολούθησε τις οδηγίες για την εγκατάσταση της Python στον υπολογιστή σου.
3. Βεβαιώσου ότι έχεις ορίσει τη μεταβλητή PATH για την Python στο σύστημά σου.

Τοπική Εκτέλεση Προγραμμάτων Python:

1. Ανοίγεις έναν κειμενογράφο (π.χ. Notepad++) ή έναν ενσωματωμένο επεξεργαστή κειμένου όπως ο IDLE ή ο PyCharm.
2. Γράφεις τον κώδικά σου.
3. Αποθηκεύεις το αρχείο με κατάληξη «.py».
4. Ανοίγεις ένα τερματικό και μεταβαίνεις στον φάκελο όπου αποθήκευσες το αρχείο Python.
5. Πληκτρολογείς python filename.py για να εκτελέσεις το πρόγραμμά σου.

Βασικοί Τύποι Δεδομένων

1. Ακέραιοι Αριθμοί: int
2. Δεκαδικοί Αριθμοί: float
3. Λογικές Τιμές: bool (παίρνουν τιμή True ή False)
4. Συμβολοσειρές: str
5. Λίστες: list
6. Πλειάδες: tuple
7. Λεξικά: dict
8. Σύνολα: set

Τελεστές

1. Αριθμητικοί Τελεστές:

Πρόσθεση: +
Αφαίρεση: -
Πολλαπλασιασμός: *
Διαίρεση: /
Υπόλοιπο: %
Ύψωση σε δύναμη: **

2. Συγκριτικοί Τελεστές:
Ίσο: ==
Όχι ίσο: !=
Μικρότερο ή Ίσο: <=
Μεγαλύτερο ή Ίσο: >=
Μικρότερο: <
Μεγαλύτερο: >
3. Λογικοί Τελεστές:
Και: *and*
Ή: *or*
Όχι: *not*

Σταθερές
Οι σταθερές είναι τιμές που δεν αλλάζουν κατά τη διάρκεια της εκτέλεσης του προγράμματος.

Παραδείγματα:
Ο αριθμός πι που είναι περίπου 3.14159.
Η ταχύτητα του φωτός στο κενό, που είναι περίπου 299,792,458 μέτρα ανά δευτερόλεπτο.

Μεταβλητές
Οι μεταβλητές είναι ονομασίες που αντιστοιχίζονται σε μια τιμή.
Παραδείγματα:
x = 5
name = "Κώστας"
is_student = True

Εκφράσεις
Οι εκφράσεις είναι συνδυασμοί τελεστών και τιμών που αποτιμούνται σε μια μοναδική τιμή.
Παραδείγματα:
*result = 2 * (3 + 4)*
is_valid = x > 10 and x < 20

Είσοδος τιμών από τον χρήστη
Με την εντολή *input()* το κείμενο που πληκτρολογεί ο χρήστης επιστρέφεται ως αλφαριθμητική τιμή. Αν θέλουμε να μετατραπεί η αλφαριθμητική τιμή που πληκτρολογούμε σε ακέραιο ή πραγματικό αριθμό, χρησιμοποιούμε τις συναρτήσεις int(), float() αντίστοιχα.
Προαιρετικά: Αν θέλουμε να εμφανιστεί ένα μήνυμα στην οθόνη πριν την είσοδο τιμής χρησιμοποιούμε την παράμετρο της input().

Παραδείγματα:
name = input("Ποιό είναι το όνομά σου;")
age = int(input("Ποιά είναι η ηλικία σου;"))

Εμφάνιση τιμών στον χρήστη
Με την εντολή print() εμφανίζονται στην οθόνη οι τιμές που δίνουμε ως παραμέτρους. Αν θέλουμε να εμφανιστούν περισσότερες από μία τιμές, τότε τις χωρίζουμε με κόμματα
Παραδείγματα:
print('Το όνομά μου είναι Κώστας')
age= 21
print('Η ηλικία μου είναι',age)
Σχόλια:
Ο χαρακτήρας # δηλώνει την έναρξη ενός σχολίου.
 # Αυτό είναι ένα σχόλιο print("Τεια σου, κόσμε!")

Προγραμματιστικές Δομές
1. Δομή επιλογής (if-elif-else):
Παραδείγματα:
A.
```
if number > 0:
    print("Θετικός αριθμός")
```
B.
```
if x > 5:
  print("Το x είναι μεγαλύτερο από 5")
elif x == 5:
  print("Το x είναι ίσο με 5")
else:
  print("Το x είναι μικρότερο από 5")
```
2. Επαναλήψεις (for, while):
Παραδείγματα:
A.
```
for i in range(6):
  print(i)
```
B.
```
  count = 0
    while count < 6:
      print(count)
      count = count + 1
```

Δομές δεδομένων στην Python
Λίστες (Lists):
Μια λίστα είναι μια συλλογή στοιχείων οποιουδήποτε τύπου δεδομένων.
```
  # Ορισμός μιας λίστας
  my_list = [1, 2, 3, 4, 5]
  # Προσθήκη στοιχείου στη λίστα
  my_list.append(6)
  # Πρόσβαση σε ένα στοιχείο με βάση τη θέση του (indexing)
  print(my_list[0]) # Εκτυπώνει: 1
  # Αφαίρεση στοιχείου από τη λίστα με βάση τη θέση του
  removed_item = my_list.pop(2)
  print(removed_item) # Εκτυπώνει: 3
```

Πλειάδες (Tuples):
Μια πλειάδα είναι μια συλλογή στοιχείων που δεν μπορούν να αλλάξουν.

```
# Ορισμός μιας πλειάδας
my_tuple = (1, 2, 3, 4, 5)
# Πρόσβαση σε ένα στοιχείο με βάση τη θέση του (indexing)
print(my_tuple[0]) # Εκτυπώνει: 1
# Προσπάθεια αλλαγής ενός στοιχείου (θα προκαλέσει σφάλμα)
my_tuple[0] = 6 # TypeError: 'tuple' object does not support item assignmen
```

Λεξικά (Dictionaries):
Ένα λεξικό είναι μια συλλογή κλειδιών-τιμών, επιτρέποντας την ανάκτηση δεδομένων μέσω του κλειδιού.

```
# Ορισμός ενός λεξικού
my_dict = {'name': 'John', 'age': 30, 'city': 'New York'}
# Πρόσβαση σε μια τιμή με βάση το κλειδί
print(my_dict['age']) # Εκτυπώνει: 30
# Προσθήκη νέου ζεύγους κλειδιού-τιμής
my_dict['job'] = 'Engineer'
# Εμφάνιση όλων των κλειδιών
print(my_dict.keys()) # Εκτυπώνει: dict_keys(['name', 'age', 'city', 'job'])
```

Σύνολα (Sets):
Ένα σύνολο είναι μια συλλογή μοναδικών στοιχείων, χωρίς διπλές καταχωρήσεις.

```
# Ορισμός ενός συνόλου
my_set = {1, 2, 3, 4, 5}
# Προσθήκη στοιχείου στο σύνολο
my_set.add(6)
# Αφαίρεση στοιχείου από το σύνολο
my_set.remove(3)
# Έλεγχος αν ένα στοιχείο υπάρχει στο σύνολο
print(2 in my_set) # Εκτυπώνει: True
```

Συναρτήσεις
Στην Python, μια συνάρτηση είναι ένα μπλοκ κώδικα που ορίζεται με ένα όνομα, μπορεί να δεχτεί είσοδο (μέσω των παραμέτρων), να εκτελέσει κάποιες ενέργειες και να επιστρέψει ένα αποτέλεσμα. Ορίζεται με τη χρήση της λέξης-κλειδιού def, ακολουθούμενο από το όνομα της συνάρτησης και τις παραμέτρους της, αν υπάρχουν, και τελικά το σώμα της συνάρτησης που περιέχει τον κώδικα που θέλουμε να εκτελεστεί. Μια συνάρτηση επιστρέφει ένα αποτέλεσμα χρησιμοποιώντας την εντολή return.
Παράδειγμα:

```
def greet(name):
  print("Γειά σου " + name + "!")
# Κλήση της συνάρτησης
greet("Αχιλλέα") # Εκτυπώνει: Γειά σου Αχιλλέα!
```

ΑΡΧΗ

3. Μικρό Εγχειρίδιο της γλώσσας προγραμματισμού Javascript

Η JavaScript είναι μια πολύ δημοφιλής γλώσσα προγραμματισμού που χρησιμοποιείται κυρίως στον παγκόσμιο ιστό.

Βασικοί Τύποι Δεδομένων
Αριθμοί: number
Συμβολοσειρές: string
Λογικές Τιμές: boolean (true ή false)
Αντικείμενα: object
Πίνακες: array
Μη Ορισμένο (undefined) και Κενό (null)

Τελεστές
Αριθμητικοί Τελεστές:
Πρόσθεση: +
Αφαίρεση: -
Πολλαπλασιασμός: *
Διαίρεση: /
Υπόλοιπο: %
Ύψωση σε δύναμη: **
Επιπλέον, υπάρχουν οι εξής τελεστές:
Αύξηση (++) και Μείωση (—): Χρησιμοποιούνται για την αύξηση ή τη μείωση ενός αριθμού κατά ένα.
Τελεστές Συντομεύσεων (+=, -=, *=, /=, %=, **=): Χρησιμοποιούνται για συντομεύσεις στις πράξεις των παραπάνω τελεστών.

Συγκριτικοί Τελεστές:
Ίσο: ==
Όχι Ίσο: !=
Μικρότερο ή Ίσο: <=
Μεγαλύτερο ή Ίσο: >=
Μικρότερο: <
Μεγαλύτερο: >
Λογικοί Τελεστές:
Και: &&
Ή: ||
Όχι: !

Σταθερές
Οι σταθερές είναι τιμές που δεν μπορούν να αλλάξουν κατά τη διάρκεια της εκτέλεσης του προγράμματος.

Παραδείγματα:
Ο αριθμός π που είναι περίπου 3.14159.
Η ταχύτητα του φωτός στο κενό, που είναι περίπου 300.000.000 μέτρα ανά δευτερόλεπτο.

Μεταβλητές
Οι μεταβλητές είναι ονομασίες που αντιστοιχίζονται σε μια τιμή.
Παραδείγματα:
var x = 5;
var name = "Κώστας";
var isStudent = true;

Εκφράσεις
Οι εκφράσεις είναι συνδυασμοί τελεστών, μεταβλητών και σταθερών που λαμβάνουν μια μοναδική τιμή.
Παραδείγματα:
*var result = 2 * (3 + 4);*
var isValid = x > 10 && x < 20;

Είσοδος τιμών από τον χρήστη
1ος Τρόπος:
Χρησιμοποιώντας τη μέθοδο prompt() μπορούμε να ζητήσουμε από τον χρήστη να εισάγει μια τιμή. Η prompt() εμφανίζει ένα παράθυρο διαλόγου όπου ο χρήστης μπορεί να εισάγει κείμενο.

```
var name = prompt("Παρακαλώ εισάγετε το όνομά σας:");
console.log("Το όνομά σας είναι: " + name);
```

2ος τρόπος:
Διάβασμα από ένα πεδίο εισαγωγής (input field): Μπορούμε να χρησιμοποιήσουμε την ιδιότητα value ενός στοιχείου εισαγωγής HTML για να διαβάσουμε την τιμή που εισάγει ο χρήστης.

```
<input type="text" id="nameInput">
<button onclick="showName()">Υποβολή</button>
<script>
function showName() {
var name = document.getElementById("nameInput").value;
console.log("Το όνομά σας είναι: " + name);
}
</script>
```

Εμφάνιση τιμών στον χρήστη
1ος τρόπος:
Χρήση της συνάρτησης alert(): Μπορουμε να χρησιμοποιήσουμε τη συνάρτηση alert() για να εμφανίσουμε ένα μήνυμα προς τον χρήστη σε ένα παράθυρο διαλόγου.

```
var message = “Καλημέρα Αχιλλέα!”;
alert(message);
```

2ος τρόπος:
Χρήση της συνάρτησης console.log(): Μπορούμε να χρησιμοποιήσουμε τη συνάρτηση console.log() για να εμφανίσουμε ένα μήνυμα στην κονσόλα του προγράμματος περιήγησης.

```
var message = “Καλημέρα Αχιλλέα!”;
console.log(message);
```

Προγραμματιστικές Δομές
Δομή επιλογής (if-else)
Παράδειγμα:

```
if (x > 5) {
console.log (“Το x είναι μεγαλύτερο από 5”);
} else {
console.log (“Το x είναι μικρότερο ή ίσο με 5”);
}
```

Επαναλήψεις (for, while)
Παράδειγμα:

```
A.
for (var i = 0; i < 5; i++) {
console.log (i);
}
B.
let i = 0;
while (i < 5) {
console.log (i);
i++;
}
```

Συναρτήσεις
Μία συνάρτηση στην Javascript ορίζεται με τη χρήση της λέξης-κλειδιού function, ακολουθούμενο από το όνομα της συνάρτησης και τις παραμέτρους της, αν υπάρχουν, και τελικά το σώμα της συνάρτησης που περιέχει τον κώδικα που θέλουμε να εκτελεστεί.
Παράδειγμα:

```
function greet(name) {
console.log(“Γειά σου, “ + name);
}
greet(“Άχιλλέα”);
```

ΑΡΧΗ

Τίτλος: **«Περιβάλλοντα και γλώσσες προγραμματισμού»**

Έκδοση: **1.5**

Ημερομηνία: **10/09/2024**

Συντονιστής ομάδας σχεδιασμού και ανάπτυξης: **Κέλλυ Σαρρή Πασχαλίδη**

Δημιουργία: **ΕΚΔΟΣΕΙΣ ΕΛΛΗΝΙΚΗ ΓΡΑΦΗ**

Το παρόν αναπτύχθηκε στο πλαίσιο της Πράξης «Συγγραφή, Αξιολόγηση και Ένταξη διδακτικών βιβλίων στο Μητρώο Διδακτικών Βιβλίων και στην Ψηφιακή Βιβλιοθήκη Διδακτικών Βιβλίων» με κωδικό ΟΠΣ (MIS) 6010165, του Προγράμματος «Ανθρώπινο Δυναμικό και Κοινωνική Συνοχή 2021-2027» που υλοποιείται από το Ινστιτούτο Εκπαιδευτικής Πολιτικής και συγχρηματοδοτείται από το Ευρωπαϊκό Κοινωνικό Ταμείο.