

# ***Hello World***

***Το πρώτο μας πρόγραμμα***

Μέσα από το **Hello World** μαθαίνει κανείς το βασικό συντακτικό και το πώς να προβάλλει μηνύματα στον χρήστη. Κατι πολύ σημαντικό για ένα πρόγραμμα.

# ***ΣΥΝΤΑΚΤΙΚΟ***

Ονομάζουμε το σύνολο από **κανόνες** που ορίζουν τους συνδυασμούς συμβόλων, που θεωρούνται σωστά δομημένα προγράμματα στη γλώσσα προγραμματισμού.

## Η εντολή Print()

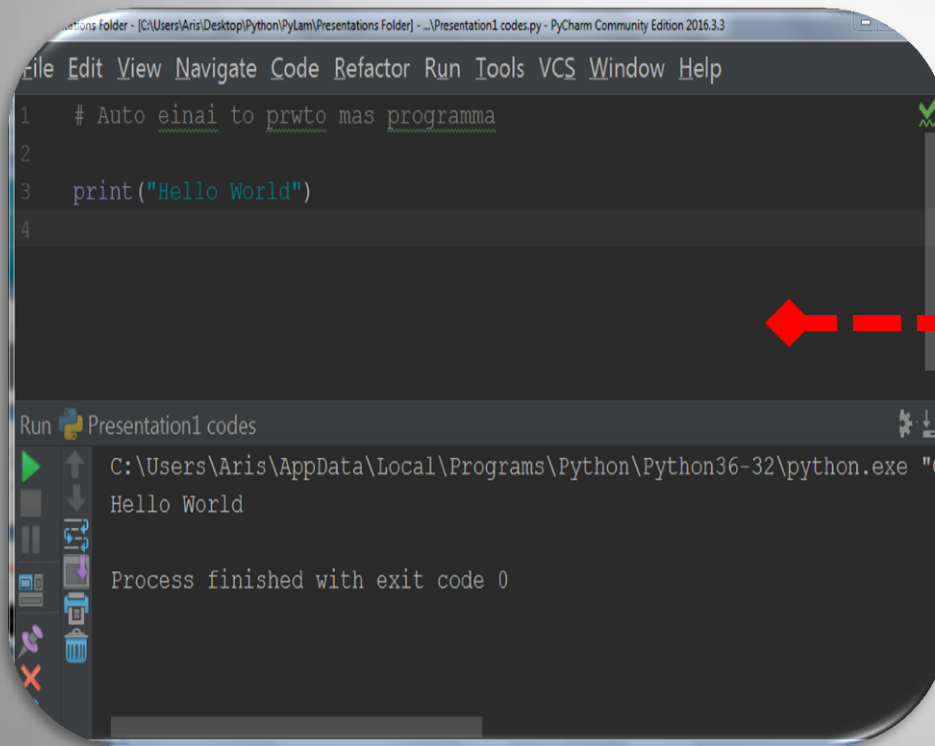
- Αν τώρα γράψουμε print(...) και μέσα αυτό που θέλουμε να τυπώσει.

π.χ. `print("Hello Python!")`

```
1  # Auto einai to prwto mas programma
2
3  print("Hello World")
4
```

# Δημιουργία του *hello\_world.py*

- Γράφοντας το παρακάτω πρόγραμμα σε κάποιον συντάκτη ή περιβάλλον προγραμματισμού, δημιουργούμε ένα python script



The screenshot shows the PyCharm IDE interface. The main editor window displays a Python script with the following code:

```
1 # Auto einai to prwto mas programma
2
3 print("Hello World")
4
```

The bottom panel shows the 'Run' output, indicating that the program executed successfully and printed 'Hello World'.



hello\_world.py

# ***Χαρακτήρες διαφυγής - 1***

- Αποτελούν χαρακτήρες τους οποίους ο υπολογιστής αντιλαμβάνεται ως κάτι παραπάνω από χαρακτήρες.
- Οι χαρακτήρες αυτοί αφορούν κάποια λειτουργία, πέραν της φυσιολογικής τους ερμηνείας.

# Χαρακτήρες διαφυγής - 2

Χαρακτήρας	Λειτουργία
\'	Χρήση single quotes
\"	Χρήση double quotes
\\	Χρήση backslash
\n	Χρήση linefeed ή newline
\t	Χρήση tab
\b	Χρήση backspace

# *Linefeed (“\n”)*

- *Linefeed* ή αλλιώς *newline*, είναι ένας ειδικός χαρακτήρας.
- Μας βοηθά να αλλάζουμε γραμμή κατά την εκτύπωση μίας σειράς χαρακτήρων.

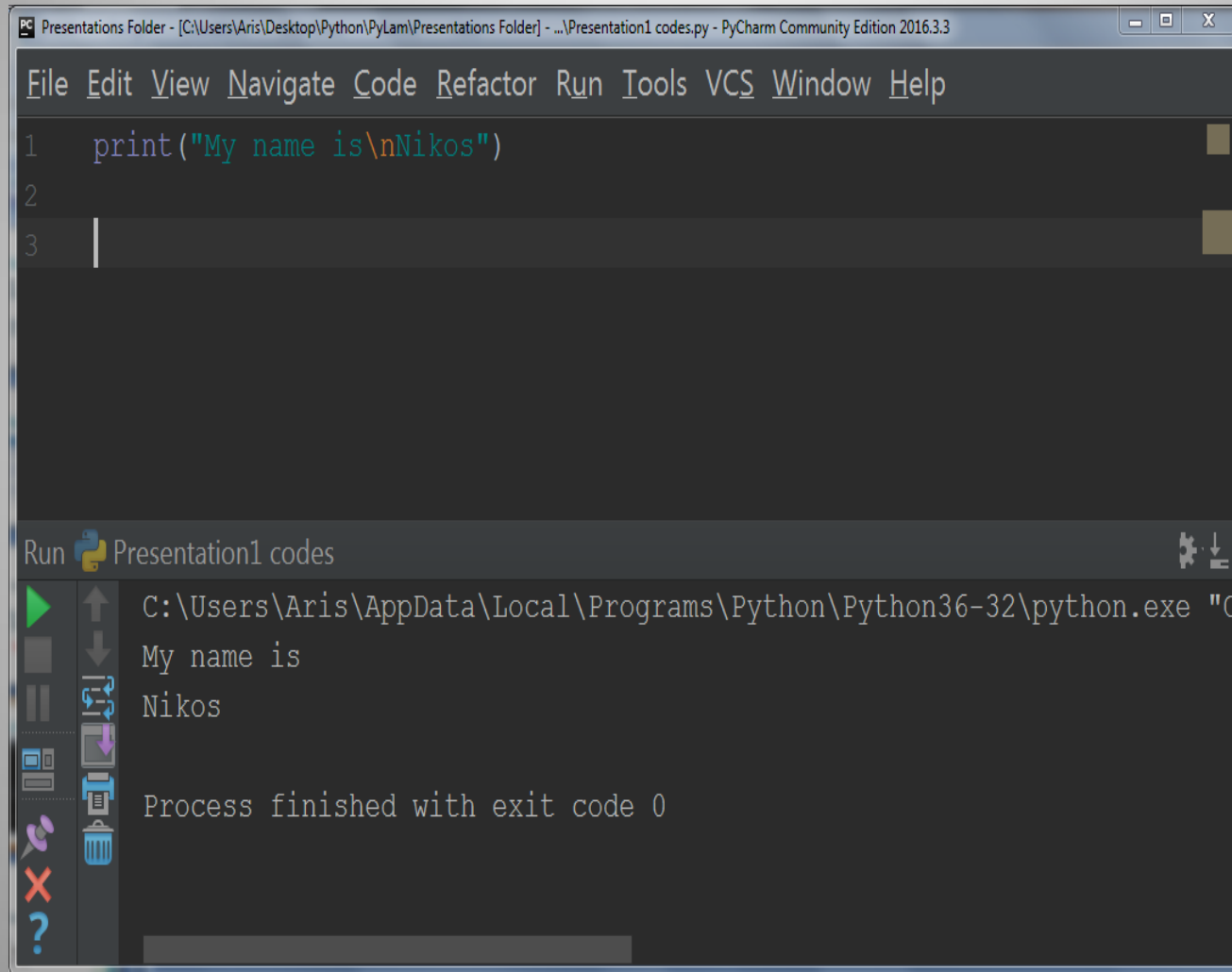
“My name is\n Nikos”



My name is  
Nikos



# Παράδειγμα του *linefeed*



The screenshot shows the PyCharm IDE interface. The top window displays a Python file named 'Presentation1 codes.py' with the following code:

```
1 print("My name is\nNikos")
2
3
```

The bottom window, titled 'Run Presentation1 codes', shows the execution output:

```
C:\Users\Aris\AppData\Local\Programs\Python\Python36-32\python.exe "C:\Users\Aris\Desktop\Python\PyLam\Presentation1 codes.py"
My name is
Nikos
Process finished with exit code 0
```

The output demonstrates the effect of the `\n` escape sequence, which inserts a line feed, resulting in the text 'My name is' and 'Nikos' being printed on separate lines.

# Άσκηση

- Γράψτε κώδικες ή κώδικα που εκτυπώνει τα παρακάτω.

1. HelloWorld
2. Hello \_\_ World
3. World “Hello” World
4. Hello  
World
5. Hello\nWorld

***Μεταβλητές***

# *Μεταβλητές της Python*

- *Θέσεις μνήμης*, οι οποίες δεσμεύονται για την αποθήκευση τιμών

Π.χ.

`x = 12`

`y = "Python"`

`z = 1.7632`

## ***Χώρος στη μνήμη***

- Οι μεταβλητές είναι λογικό να καταναλώνουν από τη μνήμη που παρέχεται.
- Όμως δεν «πιάνουν» όλες οι μεταβλητές τον ίδιο χώρο. Ανάλογα με τον **τύπο** της, η μεταβλητή πιάνει διαφορετικό χώρο.

# ***Τύποι μεταβλητών - 1***

- **Ακέραιοι (integers / int)** : Αποδεκτές τιμές είναι το σύνολο των ακεραίων.  
π.χ. 5, -5, 100, 11012
- **Κινητής υποδιαστολής (floats / float)**: Αποδεκτές τιμές είναι το σύνολο των πραγματικών.  
π.χ. 1.0, 12.4, -2.0, -10.34
- **Συμβολοσειρές (strings / str)**: αποδεκτές τιμές είναι όλα τα αλφαριθμητικά καθώς και οι ειδικοί χαρακτήρες.  
π.χ. "Nikos", "\n", "1234", "\t"

## ***Τύποι μεταβλητών - 2***

- **Λογικές (boolean)** : Αποδεκτές τιμές είναι μόνο το True και το False

Ο τύπος αυτός συναντάται κυρίως όταν υπάρχουν συνθήκες στον κώδικα μας

# ***Ο χώρος στη μνήμη***

<i><b>Τύπος μεταβλητής</b></i>	<i><b>Μέγεθος στη μνήμη</b></i>
<b>Ακέραιοι (Integers)</b>	<b>4 Bytes</b>
<b>Κινητής Υποδιαστολής (Floats)</b>	<b>4 Bytes</b>
<b>Αλφαριθμητικά (strings)</b>	<b>1 Byte</b>



# ***Παραδείγματα***

## ***Παράδειγμα 1:***

$x = 5$

$y = 3.15$

name = "Giorgos"

continue = True

## ***Παράδειγμα 2:***

Ένα άτομο:

name = "Giorgos"

age = 18

height = 1.80

# *Typecast*

- Μέσω αυτής της λειτουργίας μπορούν να μετατρέπουμε μεταβλητές του ενός τύπου σε κάποιον άλλο.

**π.χ.**

`x = 5`, είναι `int`

`y = str(x)`, πλέον είναι το `"5"`

# *User Input*

- Το *user input* είναι ο τρόπος με τον οποίο δίνουμε τιμές σε μεταβλητές μέσα από το πληκτρολόγιο

Η εντολή που μας βοηθά να το επιτύχουμε αυτό είναι η *input()*

## ***Εντολή `input()`***

Το μόνο που χρειάζεται να κάνουμε είναι να γράψουμε την μεταβλητή μας, να την εξισώσουμε με την είσοδο και τέλος να εισάγουμε τις τιμές.

Τέλος πατάμε **enter** για να συνεχίσει η εκτέλεση.

# ***Εκτύπωση μεταβλητών***

Για να προβάλλουμε το περιεχόμενο μιας μεταβλητής το μόνο που χρειάζεται είναι μια εντολή ***print()***

Αλλά, τώρα στην παρένθεση βάζουμε την μεταβλητή, της οποίας την τιμή θέλουμε να εμφανίσουμε

## *Επιπλέον χαρακτηριστικά*

- Μπορούμε να προσθέσουμε ένα αλφαριθμητικό στην παρένθεση του *input()* το οποίο θα προβάλλεται όταν τρέχει ο κώδικας!

Presentations Folder - [C:\Users\Aris\Desktop\Python\PyLam\Presentations Folder] - ...Presentation1 codes.py - PyCharm Community Edition 2016.3.3

File Edit View Navigate Code Refactor Run Tools VCS Window Help




```
1 name = input("Enter your name: ")
```



```
2
```

```
3 print(name)
```

```
4
```

```
5
```

Run  Presentation1 codes  

  C:\Users\Aris\AppData\Local\Programs\Python\Python36-32\python.exe "C

Enter your name: Taxiarchis

Taxiarchis

Process finished with exit code 0

# ***Παραδείγματα***

- Για ένα άτομο όπως το ορίσαμε πριν θα είναι

```
name = input("Enter your name:")  
age = int(input("Enter your age:"))  
height = float(input("Enter your  
height:"))
```



# *Γιατί `int(input())`??*

Επειδή ό,τι διαβάζεται από το πληκτρολόγιο με την εντολή ***input*** μετατρέπεται σε μια συμβολοσειρά, ο προγραμματιστής οφείλει να “πει” στον υπολογιστή τι τύπου μεταβλητή είναι αυτή που ζητάει.

Εδώ λοιπόν έρχεται το ***typecasting***, το οποίο μας βοηθά να επιτύχουμε αυτήν την μετατροπή.

**Βασικές πράξεις** αποκαλούμε τις διαδικασίες που αποτελούνται από μεταβλητές και σύμβολα (τελεστές) και παράγουν **νέες** τιμές

# ***Παραδείγματα χρήσης***

- Αριθμητικές πράξεις
- Εκχώρηση τιμών σε μεταβλητές
- Έλεγχος συνθηκών
- Άλγεβρα Boole
- Χειρισμός δυαδικών

# ***Τελεστές***

- Είναι τα σύμβολα με τα οποία εκτελούμε τις βασικές πράξεις
- Ανάλογα με το είδος της πράξης χωρίζονται σε κατηγορίες

# ***Κατηγορίες τελεστών***

- **Εκχώρησης**
- **Αριθμητικοί**
- **Σχεσιακοί**
- **Λογικοί**

*Τελεστής εκχώρησης (=)*

*μεταβλητή = τιμή*

# Αριθμητικοί τελεστές

Τελεστές	Πράξη	Παραδείγματα
+	Πρόσθεση	$y = a + b$
-	Αφαίρεση	$y = a - b$
*	Πολλαπλασιασμός	$y = a * b$
/	Διαίρεση	$y = a / b$
**	Ύψωση σε δύναμη	$y = b ** 2$
//	Ακέραια διαίρεση	$y = a // b$
%	Υπόλοιπο ακέραιας διαίρεσης	$y = a \% b$

# ***Αριθμητικές παραστάσεις***

Αποτελεί ένα σύνολο μεταβλητών και αριθμών, όπου τα στοιχεία του συνόλου συνδέονται μεταξύ τους με αριθμητικούς τελεστές.



# Παραδείγματα

Αν  $a = 5$ ,  $b = 6$ , ποια είναι τα αποτελέσματα των παρακάτω αριθμητικών παραστάσεων;

- $a + b$
- $a - b**b$
- $a * (b/a)$
- $a**b$
- $(a \% b) + (a // b)$

# ***Σχεσιακοί τελεστές - 1***

- Εκφράζουμε ισότητες, ανισότητες και ανισοϊσότητες
- Τοποθετούνται ανάμεσα σε μεταβλητές και το αποτέλεσμα είναι Αληθές ή Ψευδές

# Σχεσιακοί τελεστές - 2

Τελεστές	Σχέση	Παραδείγματα (a = 12, b = 13)
==	Ισότητα	a == b : <b>False</b>
!=	Ανισότητα	a != b : <b>True</b>
>	Μεγαλύτερο	a > b: <b>False</b>
<	Μικρότερο	a < b: <b>True</b>
>=	Μεγαλύτερο ή Ίσο	a >= b: <b>False</b>
<=	Μικρότερο ή Ίσο	a <= b: <b>True</b>

# Παρατηρήσεις

- **Προσοχή!!** Το `==` είναι διαφορετικό από τον τελεστή εκχώρησης `=`
- Στις ανισοϊσότητες οι τελεστές έχουν αυστηρώς την μορφή `<=`, `>=` και ποτέ `=<` και `=>`
- Δεν μπορούμε να εισάγουμε τρεις μεταβλητές σε κάποια σχέση

# ***Λογικοί τελεστές - 1***

- Χρησιμοποιούνται σε λογικές πράξεις με λογικές μεταβλητές
- Οι λογικές πράξεις έχουν αποτελέσματα Αληθές ή Ψευδές (True ή False)

## Λογικοί τελεστές - 2

Τελεστές	Σχέση	Παραδείγματα (a = 12, b = 13)
and	και	(a == b) and (a < b) : <b>False</b>
or	ή	(a == b) or (a < b) : <b>True</b>
not	όχι	not(a < b): <b>False</b>

# *Ο Πίνακας Αληθείας*

a	b	and	or	not (a)	not (b)
T	T	T	T	F	F
T	F	F	T	F	T
F	T	F	T	T	F
F	F	F	F	T	T

# Παραδείγματα

Έστω οι μεταβλητές  $a = \text{True}$ ,  $b = \text{False}$ . Βρείτε τα αποτελέσματα των παρακάτω λογικών παραστάσεων.

- $a \text{ and } b$      $\# \text{ False}$
- $a \text{ or } b$      $\# \text{ True}$
- $\text{not } a$      $\# \text{ False}$
- $(a \text{ and } b) \text{ and } ((\text{not } a) \text{ or } b)$      $\# \text{ False}$
- $\text{not}(a \text{ or } b) \text{ or } \text{not}(a \text{ and } b)$      $\# \text{ True}$
- $(a \text{ or } (\text{not } b)) \text{ and } ((\text{not } a) \text{ or } b)$      $\# \text{ False}$



# Τελεστές *Super-wow*

Τελεστής	Επεξήγηση
<b><code>+=</code></b>	<b><code>i += 1</code> <math>\rightarrow</math> <code>i = i + 1</code></b>
<b><code>-=</code></b>	<b><code>i -= 1</code> <math>\rightarrow</math> <code>i = i - 1</code></b>
<b><code>*=</code></b>	<b><code>i *= 1</code> <math>\rightarrow</math> <code>i = i * 1</code></b>
<b><code>/=</code></b>	<b><code>i /= 1</code> <math>\rightarrow</math> <code>i = i / 1</code></b>
<b><code>**=</code></b>	<b><code>i **= 1</code> <math>\rightarrow</math> <code>i = i ** 1</code></b>
<b><code>//=</code></b>	<b><code>i //= 1</code> <math>\rightarrow</math> <code>i = i // 1</code></b>
<b><code>%=</code></b>	<b><code>i %= 1</code> <math>\rightarrow</math> <code>i = i % 1</code></b>

# *Πίνακας Προτεραιοτήτων*

<b>** , ()</b>
<b>*, /, %, //</b>
<b>+, -</b>
<b>&lt;=, &gt;=, &lt;, &gt;</b>
<b>==, !=</b>
<b>=, +=, -=, *=, /=, **=, //=, %=</b>
<b>not, or, and</b>