

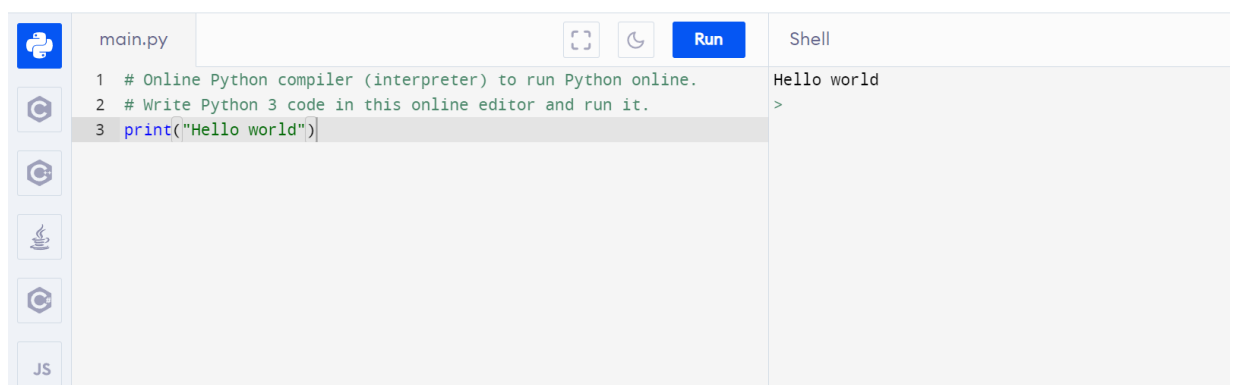
1. What is Python & history

Guido van Rossum created Python in the late 1980s. Python is a programming language that allows you to work faster and more effectively with computers. Python is a popular program.

2. Install Python and install an IDE such as Pycharm



3. Print your first code (hello world)



4. What are Variables and show some simple examples in your IDE

-A variable in a Python program provides data to the computer for processing.

<div>Execute Share main.py STDIN</div> <pre>1 # Hello World program in Python 2 3 a=50 4 print (a) 5</pre>	<div>Result</div> <pre>\$python main.py 50</pre>
--	--

5. How do you swap variables + examples

<div>Execute Share main.py STDIN</div> <pre>1 # Hello World program in Python 2 3 a=4 4 b=5 5 temp=a #4 6 a=b #5 7 b= temp #4 8 print(a) 9 print(b) 10 11</pre>	<div>Result</div> <pre>\$python main.py 5 4</pre>
---	---

6. Variable types

There are five different sorts of variables:

- ❖ Numbers
- ❖ Strings
- ❖ Lists
- ❖ Tuples
- ❖ Dictionaries

7. Variable number arithmetic operators

7 arithmetic operators

- ❖ Addition +
- ❖ subtraction -
- ❖ multiplication *
- ❖ division /

- ❖ modulus %
- ❖ exponent **
- ❖ floor division //

```

1  # Hello World program in Python
2  a=1
3  b=5
4  c=b
5
6  (c+a)+b
7
8
9

```

Part 2:

1. if clause

Execute	Share	main.py	STDIN	Result
<pre> 1 # Hello World program in Python 2 a=10 3 b=15 4 if a<b: 5 print('a is less than b') 6 print('not sure if a is less than b') 7 8 </pre>				<pre> \$python main.py a is less than b not sure if a is less than b </pre>

2. else

Execute	Share	main.py	STDIN	Result
<pre> 1 # Hello World program in Python 2 a=3 3 b=4 4 if a<b: 5 print('a is less than b') 6 else: 7 print('a is Not less than b') 8 9 10 </pre>				<pre> \$python main.py a is less than b </pre>

3. equal to (==)

Execute	Share	main.py	STDIN	Result
<pre> 1 # 2 a=11 3 b=9 4 if a<b: 5 print('a is less than b') 6 elif a==b: 7 print('a is equal to b') 8 else: 9 print('a is greater than b') 10 11 12 </pre>				<pre> \$python main.py a is greater than b </pre>

4. BMI calculator

Is the finalResult 48? Yes, well done!

The final result is odd. Hrm.

```
let finalResult;
let evenOddResult;

// Add your code here
var num1= 10;
var num2= 2;
var num3= 12;
var num4 =16;
num5= num1+num2
num6= num4-num3
finalResult=num5*num6

// Don't edit the code below here!

section.innerHTML = '';
let para1 = document.createElement('p');
let finalResultCheck = finalResult === 48 ? `Yes, well done!` : `No, it is ${ finalResult }`;
para1.textContent = `Is the finalResult 48? ${ finalResultCheck }`;
```

5. Functions

```
1  #!/usr/bin/python
2
3  Def functionname (): // use def
4  Def functionname():
5      print ("hello")
6      print ("hello hi 2")
7
8  print ('this is outside the function')
```

6. BMI Calculator (using functions)

```
Execute | Share | main.py | STDIN
1  #!/usr/bin/python
2
3  name="sara"
4
5  height_m=1.69
6
7  weight_kg=65
8
9
10
11 bmi = weight_kg / (height_m * height_m)
12
13 print ('bmi: ')
14
15 print (bmi)
16
17
18
19 if bmi < 25:
20     print (name)
21     print ('is not overweight')
22
23 else:
24     print (name)
25     print ("is overweight")
26
27
28
29
```

Part 3

Loops -

1. while

In the Python programming language, a **while loop** statement executes a target statement repeatedly as long as a specific condition is true.

2. for

Execute Share main.py STDIN	Result
<pre>1 #!/usr/bin/python 2 3 fruits = ['pear', 'pineapple', 'grape'] 4 for index in range(len(fruits)): 5 print 'Current fruit :', fruits[index] 6 7 print "hello!"</pre>	<pre>\$python main.py Current fruit : pear Current fruit : pineapple Current fruit : grape hello!</pre>

3. nested

You can use one or more loops inside any another while, for or do..while loop

Functions -

1. Creating a function

```
def functionname( parameters ):  
    "function_docstring"  
    function_suite  
    return [expression]
```

2. Printme() function

```
1  #!/usr/bin/python  
2  
3  # Function definition is here  
4  def printme( str ):  
5      "This prints a passed string into this function"  
6      print str  
7      return;  
8  
9  # Now you can call printme function  
10 printme( str = "Hello Sara")
```

\$python main.py
Hello Sara

3. Keywords argument function

The function calls are linked to keyword arguments. When you utilize keyword arguments in a function call, the caller uses the parameter name to identify the parameters.

Lists -

1. Creating lists

Lists in Python are useful if you want to create a list or a category.

- ❖ you have to use square brackets
- ❖ a comma

example:

```
list1 = ['physics', 'chemistry', 1997, 2000];  
list2 = [1, 2, 3, 4, 5 ];  
list3 = ["a", "b", "c", "d"]
```

2. Accessing values in a list (Tip: Printing specific values from a list using square brackets for slicing along with the index or indices to obtain value available at that index.

Example

```
List1_friends=['sara','maria','oona']
```

```
List2_fruits=['apple','pineapple','orange']
```

3. Basic List operators

Python Expression	Results	Description
<code>len([1, 2, 3])</code>	3	Length
<code>[1, 2, 3] + [4, 5, 6]</code>	<code>[1, 2, 3, 4, 5, 6]</code>	Concatenation
<code>['Hi!'] * 4</code>	<code>['Hi!', 'Hi!', 'Hi!', 'Hi!']</code>	Repetition
<code>3 in [1, 2, 3]</code>	True	Membership
<code>for x in [1, 2, 3]: print x,</code>	1 2 3	Iteration

References:

https://www.tutorialspoint.com/python/python_lists.htm Accessed 29.11.2021

https://www.tutorialspoint.com/execute_python_online.php Accessed 22.11.2021

<https://www.youtube.com/watch?v=iS3PUCcCuVM> Accessed 23.11.2021

https://www.youtube.com/watch?v=5JsX_B885Fc Accessed 23.11.2021