



Python Basics



Session-1



Python Overview Video





Before breaking the jug



Things to do...

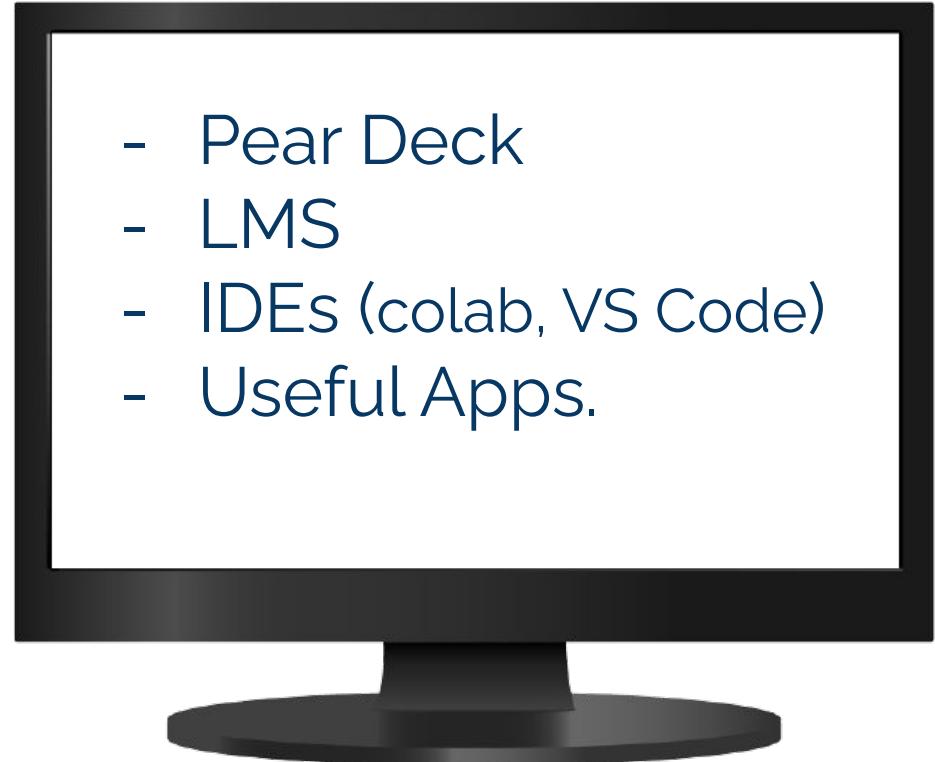


Hands off the keyboards...





Using of two screens





General Information about Python





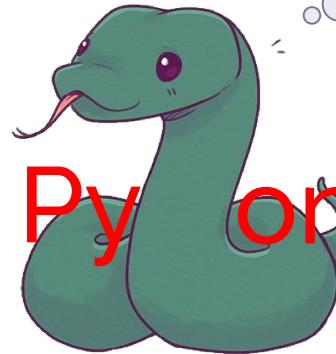
Table of Contents

- ▶ What is Python?
- ▶ Historical Development of Python
- ▶ Review of Tools & Installations
- ▶ First Program 'Hello World!'
- ▶ Matter of Quotes



1

What is Python?



A snake has
never been so
cute.



What did you learn from Pre-class materials?

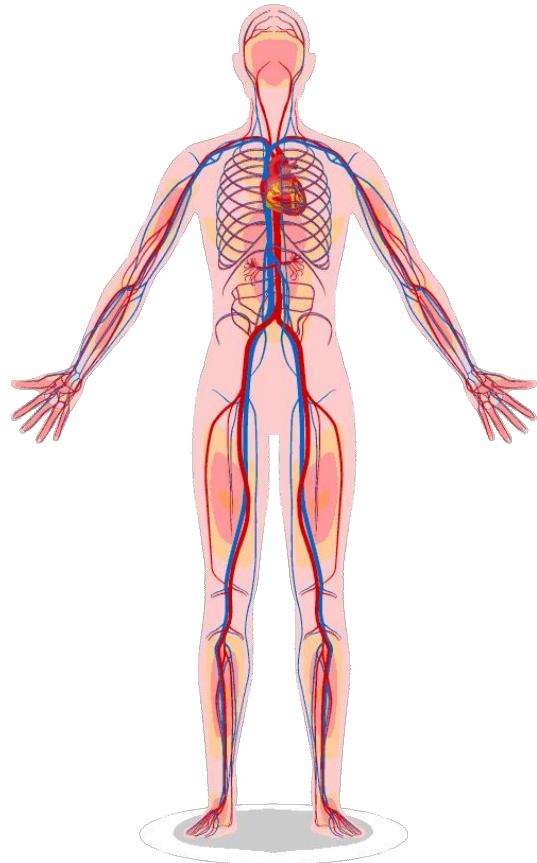
Write down three things..

- *
- *
- *



Students, write your response!

What is Python?



Mr. IT

What is Python? (review)



Less code



Pre-built libraries



Ease of learning



Platform
Independent

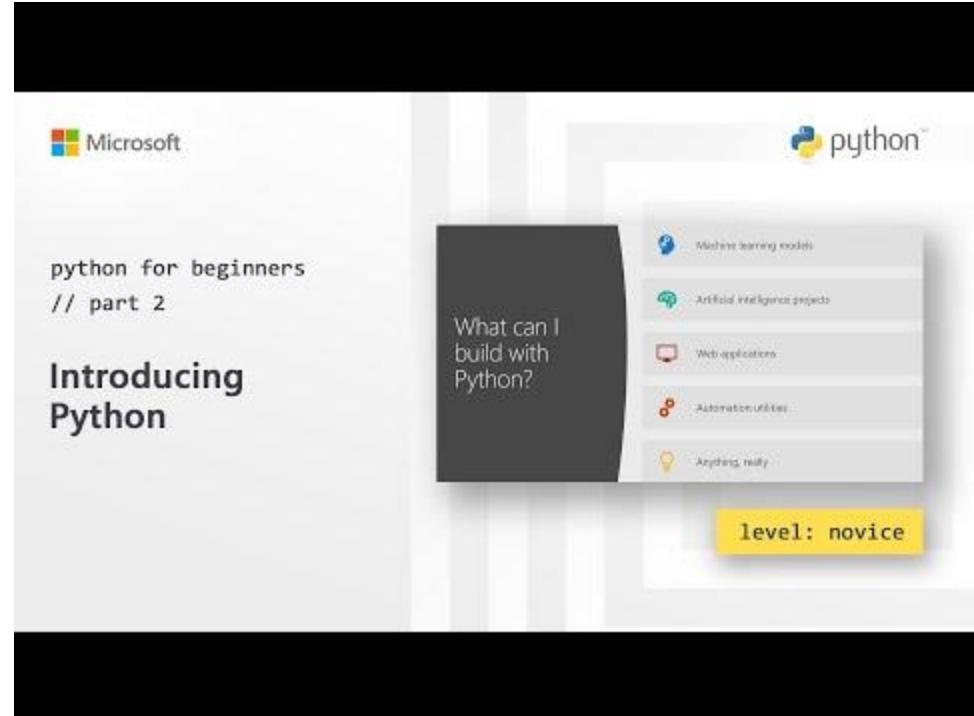


Massive Community
Support



What is Python?

- Easy to **learn**
- Easy to **use**
- Easy to **run**
- Easy to **read**
- Easy to **develop**
- Easy to **teach..**



'Microsoft Developers' on Youtube :
-Introducing Python (*duration: 3 min 9 sec*)

What do you think about
World-class companies
and institutions that built
their technical
infrastructure using
Python?



Pear Deck

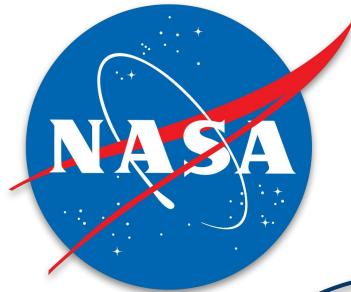


Students, write your response!

Pear Deck Interactive Slide
Do not remove this bar



What is Python?



Google

amazon.com®

N

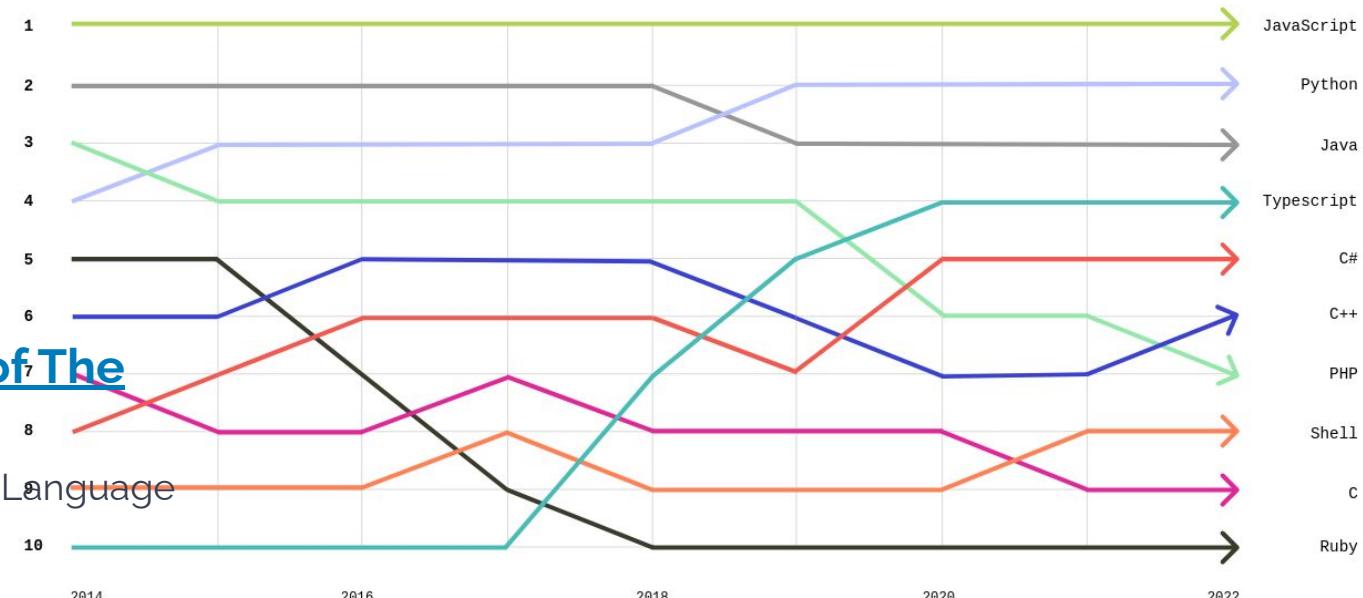
Why Python?

Some indexes showing the position of Python among the other programming languages.



Why Python?

Some indexes showing the position of Python among the other programming languages.



2022 Github's State of The Octoverse Report :

-The Second Most Popular Language



Why Python?

Some indexes showing the position of Python among the other programming languages.

2023 Jan Tiobes.com's
Index :

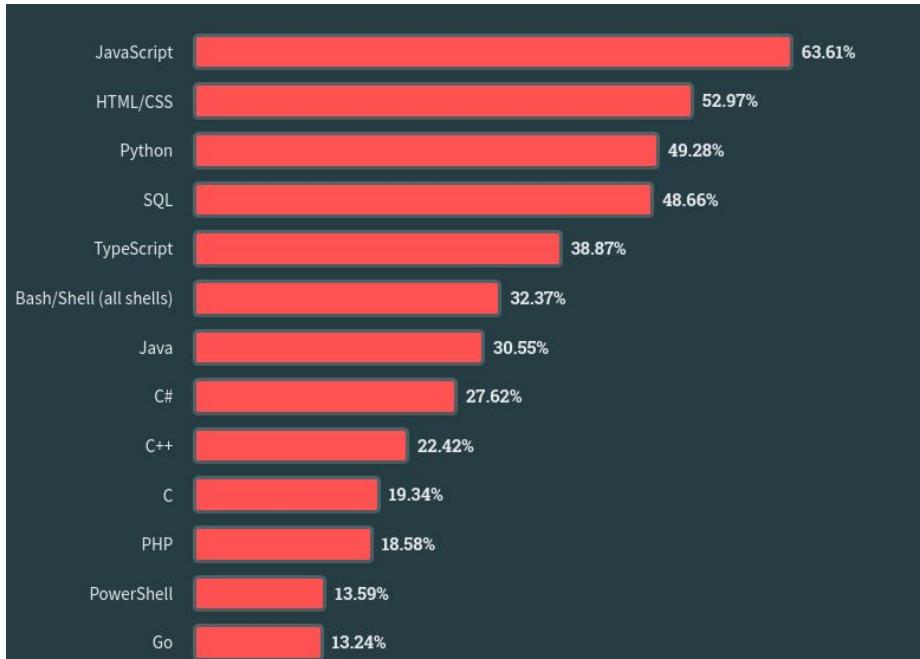
-The fastest growing language

Programming Language	2023	2018	2013	2008	2003	1998	1993
Python	1	4	8	7	13	25	17
C	2	2	1	2	2	1	1
Java	3	1	2	1	1	18	-
C++	4	3	4	4	3	2	2
C#	5	5	5	8	10	-	-
Visual Basic	6	15	-	-	-	-	-
JavaScript	7	7	11	9	8	22	-
Assembly language	8	12	-	-	-	-	-
SQL	9	251	-	-	7	-	-
PHP	10	8	6	5	6	-	-
Objective-C	18	18	3	46	49	-	-
Ada	27	30	17	18	15	8	7
Lisp	29	31	13	16	14	7	4
Pascal	211	140	15	20	99	12	3



Why Python?

Some indexes showing the position of Python among the other programming languages.



A summary slide for the 2023 Developer Survey. It features a logo of a stylized book or fan icon, the year '2023', and the title 'Developer Survey'. Below the title is a paragraph of text about the survey's scope. At the bottom are two blue buttons: 'Read the overview →' and 'Methodology →'. A black arrow points from the text above to this slide.

In May 2023 over 90,000 developers responded to our annual survey about how they learn and level up, which tools they're using, and which ones they want.

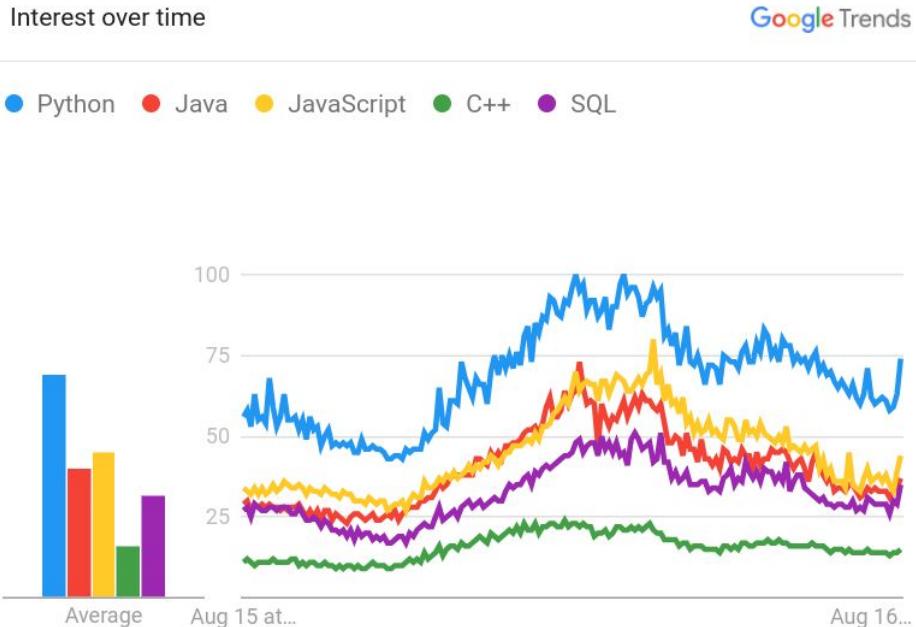
[Read the overview →](#) [Methodology →](#)

The second most popular language.



Why Python?

Some indexes showing the position of Python among the other programming languages.



2023 'trends.google.com' Report :

-The Most Popular Searched Term Among all Programming Languages



2

Historical Development of Python

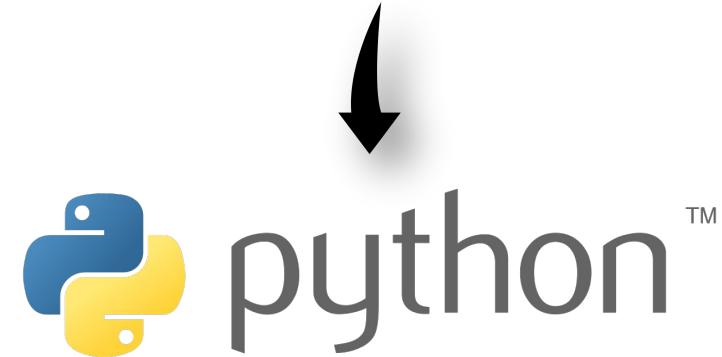
► Historical Development of Python



Guido van Rossum *(duration :4 min)*



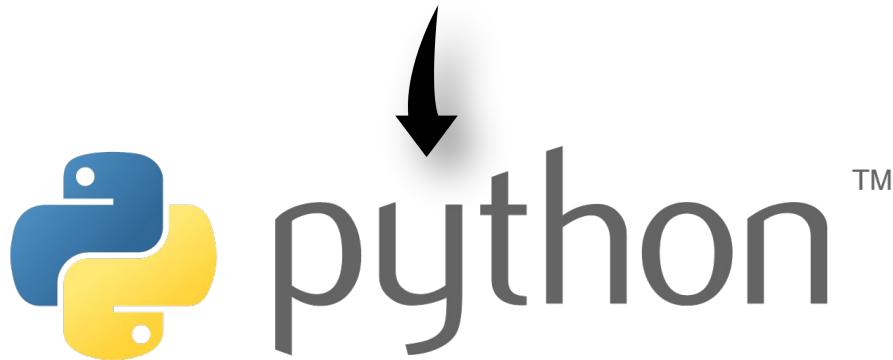
The logo



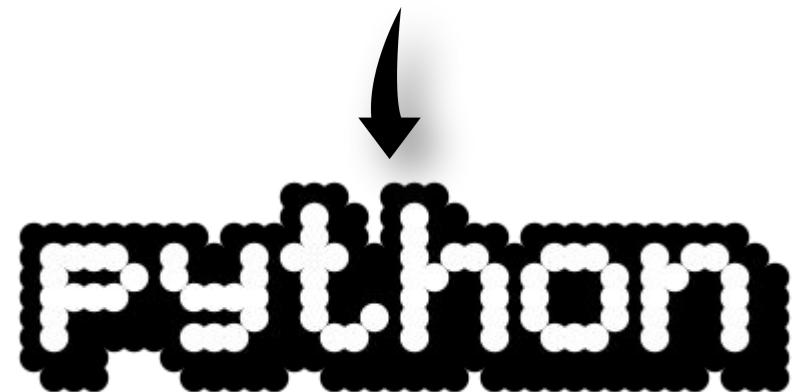


Python Logo

Python logo, used between
2006s–Today (Tim Parkin)



Python logo, used between
1990s–2006 ((Guido))





3

Review of Tools & Installations

Review of Tools & Installations

How and where to run your Python codes?

The image shows two side-by-side screenshots of Python code execution environments.

Python Playground: A light blue interface with a Python logo icon. It greets the user "Hi Joseph" and provides instructions: "you can write your code using the editor below. Once you write the code, click the run button to see the result." The code editor contains the following Python code:

```
1 print('hello world')
2
```

A green "Run" button is visible. Below the editor is an "Output" section showing the result: "hello world".

Google Colab Jupyter Notebook: A white interface with a "CO" logo icon. The title bar says "Untitled". The menu bar includes File, Edit, View, Insert, Runtime, Tools, Help, and a dropdown for "All c...". The toolbar includes Comment, Share, and settings. The sidebar shows "+ Code" and "+ Text" buttons, and a file tree with a single folder icon. The main content area is titled "Google Colab Jupyter Notebook". It displays the following Python code and output:

```
[1] print ("hello")
⇒ hello
```

At the bottom of the Colab interface is a toolbar with various icons for file operations.

I have no problem with using Playground and I liked it.

True

False



Pear Deck



Students choose an option

Review of Tools & Installations

-The current version is → **Python 3.11**

-The official web-site : www.python.org

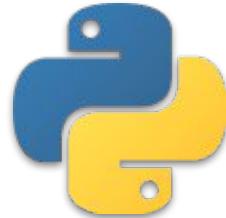
The screenshot shows the Python.org homepage with a dark blue header. The Python logo is on the left, followed by the word "python" in lowercase. On the right of the header are buttons for "Donate", a search bar with a magnifying glass icon, and "GO". Below the header is a navigation menu with tabs: "About", "Downloads", "Documentation", "Community", "Success Stories", "News", and "Events". The "Downloads" tab is currently selected. A sidebar on the left contains a Python code snippet for calculating Fibonacci numbers:

```
# Python 3: Fibonaci sequence
>>> def fib(n):
>>>     a, b = 0, 1
>>>     while a < n:
>>>         print(a, end=' ')
>>>         a, b = b, a+b
>>>     print()
>>> fib(1000)
0 1 1 2 3 5 8
987
```

The main content area has a light blue background. It features a section titled "Python Source" with a button for "Python 3.11.4". Below this, text states: "Not the OS you are looking for? Python can be used on many operating systems and environments." and "View the full list of downloads." At the bottom of the page, a summary statement reads: "Python is a programming language that lets you work quickly and integrate systems more effectively." followed by a link "» Learn More".



First Steps into Coding



Stretch Break!

Let's take 3 minutes to stretch

- *stretch your neck*



Students, follow the instructions on the slide

Introduction





4

First Program 'hello world'



Hello World!

"Hello World!" Program in Python

```
print("Hello World !")
```

"Hello World!" Program in C++

```
#include <iostream>
using namespace std ;
int main( )
{
    cout << "Hello World !";
    return 0 ;
}
```

"Hello World!" Program in C

```
#include <stdio.h>
int main( )
{
    printf("Hello World !");
    return 0 ;
}
```

"Hello World!" Program in Java

```
public class HelloWorld {
    public static void main(Strings[ ] args) {
        System.out.println("Hello World !");
    }
}
```

First Program for 'Hello World!'

- ▶ Writing a text is **quite simple** in Python.

```
print('Hello World!')
```

First Program for 'Being a Good Person'

- ▶ Writing a text is **quite simple** in Python.

```
print('Hello World!')
```

Hello World!

First Program for 'Being a Good Person'

- ▶ Writing a text is **quite simple** in Python.

```
print('Hello World!')
```

Hello World!

- ▶ Just type your text enclosed by **quotes** in parentheses.



5

Matter of Quotes

Matter of Quotes

What we wrote is a **string datatype**. Strings are always in **quotes**

There are basically two types of quotes we use in Python. **Single** or **double** quotes. Both are the same but we should use them in the correct way.

Single



```
print('Hello World!')
```

Double



```
print("Hello World!")
```

Matter of Quotes



Choose one of them and stick to it.



```
print('Hello World!')  
print("Clarusway School!")  
print("I'm happy to learn!")
```



```
print("Hello World!")  
print('Clarusway School!')  
print('I'm happy to learn!')
```

Matter of Quotes



The alternative valid use of quotes.

```
print('''Lorem ipsum dolor sit amet,  
fusce ut quisque neque donec,  
massa metus amet, luctus inceptos.''')
```

```
print(""""Praesent a, mi mattis velit metus,  
accumsan adipiscing ipsum sit justo  
penatibus, amet mauris non tempus justo.""""")
```

Triple use
of '**single**'
and
"double"
quotes

Matter of Quotes



The alternative valid use of quotes.

Output1:

 Lorem ipsum dolor sit amet,
 fusce ut quisque neque donec,
 massa metus amet, luctus inceptos.

Output2:

 Praesent a, mi mattis velit metus,
 accumsan adipiscing ipsum sit justo
 penatibus,
 amet mauris non tempus justo.

Triple use of
'single and double'
quotes

Matter of Quotes



What is the difference of these two syntaxes?

```
print('3.14')
print(3.14)
```



JSWAY[©]
REINVENT YOURSELF

Students, write your response!

Matter of Quotes



What is the difference of these two syntaxes?

```
print('3.14')  
print(3.14)
```

The outputs are the same but
quotes make data **string** type.

```
3.14  
3.14
```

Matter of Quotes



What is the output?

```
print('''We should have enough time for our family''')
```



JSWAY[©]
REINVENT YOURSELF

Students, write your response!

Pear Deck Interactive Slide
Do not remove this bar

Matter of Quotes



What is the output?

```
print('' ''We should have enough time for our family'' '')
```

'We should have enough time for our family"

```
print('We should have enough time for our family')
```

This code prints the statement inside the `print()` function.

True

False



Students choose an option

Matter of Quotes



```
print('We should have enough time for our family')
```

output

```
File "", line 1
  print('We should have enough time for our family')
                                         ^
SyntaxError: EOL while scanning string literal
```

Matter of Quotes



Multiple lines of codes.

```
print('first line')
print()
print('''third line''')
```

print() prints
an empty line

Matter of Quotes



Multiple lines of codes.

```
print('first line')
print()
print('''third line''')
```

first line

third line

print() prints
an empty line

These two codes give an empty line each.

```
print()  
print('')
```

True

False



Students choose an option

These two codes give an empty line each.

```
print()  
print('')'
```

True

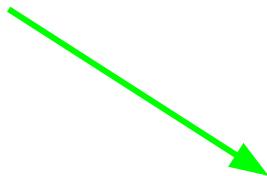
False



Students choose an option

Escape Characters

Escape characters used with `'print()'`.



Addition to:

`'sep='` : To put separators between the characters in the output message.

`'end='` : To edit the output message end.

`'file='` : Writes the output to the file.

Escape Sequence	Meaning
<code>\newline</code>	Backslash and newline ignored
<code>\\"</code>	Backslash (\)
<code>\'</code>	Single quote ('')
<code>\"</code>	Double quote ("")
<code>\a</code>	ASCII Bell (BEL)
<code>\b</code>	ASCII Backspace (BS)
<code>\f</code>	ASCII Formfeed (FF)
<code>\n</code>	ASCII Linefeed (LF)
<code>\r</code>	ASCII Carriage Return (CR)
<code>\t</code>	ASCII Horizontal Tab (TAB)
<code>\v</code>	ASCII Vertical Tab (VT)
<code>\ooo</code>	Character with octal value <i>ooo</i>
<code>\xhh</code>	Character with hex value <i>hh</i>

Escape Characters

Escape characters used with `'print()'`.

```
print("Escape\tCharacters")
```

Escape Characters

```
print("Escape\nCharacters")
```

Escape
Characters

```
print("Escape\"Char\"acters")
```

Escape"Char"acters

```
print('Escape\'Char\'acters')
```

Escape'Char'acters



PEP 8 Conventions

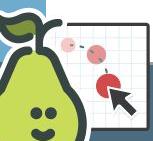




Table of Contents

- ▶ What is PEP 8?
- ▶ Some Important PEP 8 Rules

How was pre-class content ?



Students, drag the icon!

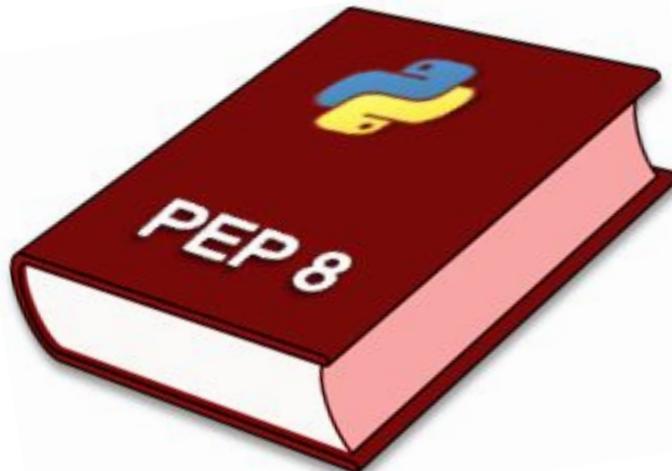


What is PEP 8?

PEP 8



Python Enhancement Proposal

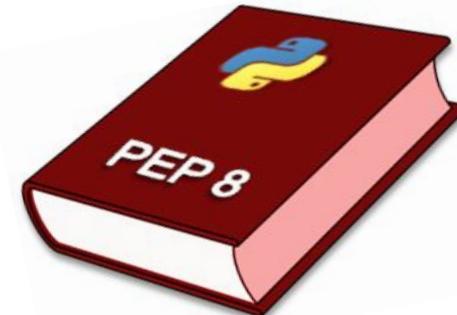




What is PEP 8?

PEP 8 is a style guide about consistency.

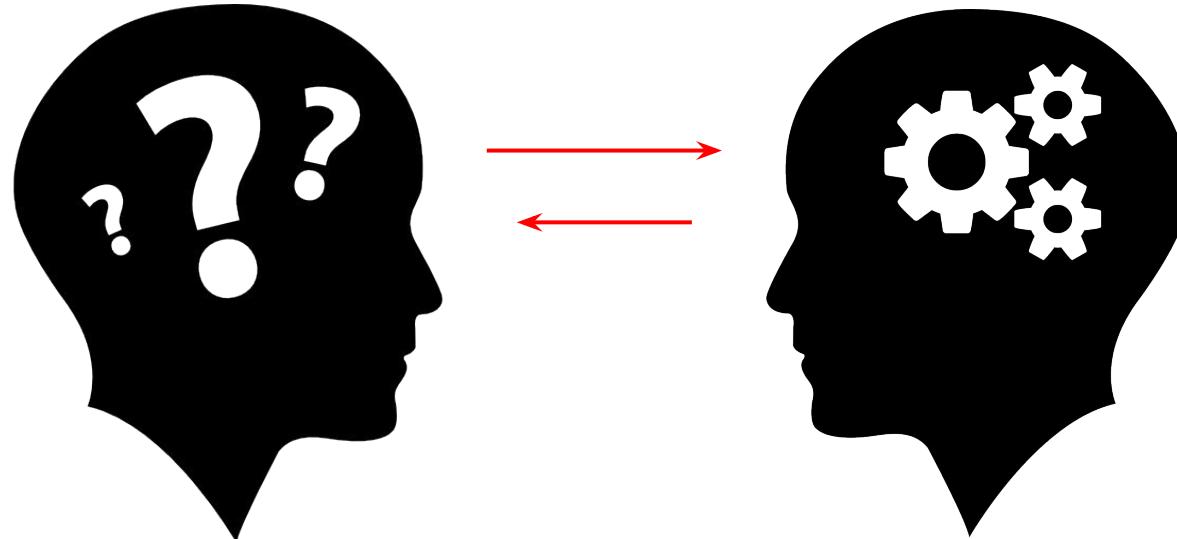
- ▶ Consistency with this style guide is **important**.
- ▶ Consistency within a project is **more important**.
- ▶ Consistency within one module or function is the **most important coding aspect**.





What is PEP 8?

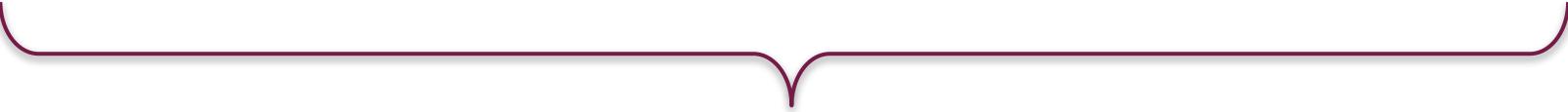
COMMUNICATE with each other!



Some Important PEP8 Rules

- ▶ Writing a text is **quite simple** in Python.

```
print('being a good person')
```



Limit the code lines to a maximum **79** characters

Some Important PEP8 Rules



```
"""This is the example docstrings.
```

```
'Comments or docstrings' are not printed on the  
screen.
```

```
"""
```

```
#being a good person ...
```

For docstrings or comments, the line length should be limited to **72** characters.



Some Important PEP8 Rules

- Avoid extraneous **spaces** in situations such as:

Immediately inside parentheses, brackets or braces :

YES : `spam(meat[1], {milk: 2})` , NO : `spam(meat[1], { milk: 2 })`

Between a trailing comma and a following close parenthesis :

YES : `df[0,] or foo = (2,)` , NO : `df[0,] or foo = (2,)`

Immediately before a comma, semicolon, or colon :

YES : `if y == 3: print x, y; x, y = y, x` , NO : `if y == 3 : print x , y ; x , y = y , x`

Immediately before the open parenthesis that starts the argument list of a function call:

YES : `print('peace')` , NO : `print ('peace')`

Some Important PEP8 Rules

- More than one space around an assignment (or other) operator to align it with another:

YES	NO
x = 3	x = 3
y = 4	y = 4
long_vars = 5	long_vars = 5

Some Important PEP8 Rules



- ▶ Always surround these operators with a single space on either side. Such as:

```
=, +=, ==, <, >, >=, in, not in, is, and, or, not
```

- ▶ Failure to follow the basic rules of PEP 8 **does not make** your program **wrong** or **unable to work**.



Don't ask '**Why**' regarding the conventional rules.





Comments and Docstrings



Introduction



I can't
remember
why I typed
these codes...

$$\begin{aligned}x &= 10 \\y &= 5 \\a &= (x / y) * 25 \\&\text{print}(a ** 2)\end{aligned}$$
$$\begin{aligned}x &= 10 \\y &= 5 \\a &= (x / y) * 100 \\&\text{print}(a)\end{aligned}$$


Comments

- ▶ **Comments** are used to explain code when the basic code itself isn't clear.

#



The '**Hash**' character makes the lines comment.



Comments

▶ **Single-line Comments :**

```
# This is a single line comment
```



Comments

► Single-line Comments :

```
# This is a single line comment
```

What is the output?

► Inline Comments :

```
print('hello') # This is an inline comment
```

two spaces

one space



These spacing principles are just PEP8 conventional rules.



USWY[®]
Students, write your response!

REINVENT YOURSELF

Pear Deck Interactive Slide
Do not remove this bar



Comments

► Single-line Comments :

```
# This is a single line comment
```

► Inline Comments :

```
print('hello') # This is an inline comment
```

two spaces

one space



These spacing principles are just PEP8 conventional rules.

```
hello
```



Comments

► **Multi-line Comments :**

```
print('hello')
# First multi-line comment
# Second multi-line comment
# Third multi-line comment
```

What will be the output ?



USWY[®]
Students, write your response!

REINVENT YOURSELF

Pear Deck Interactive Slide
Do not remove this bar



Comments

► **Multi-line Comments :**

```
print('hello')
# First multi-line comment
# Second multi-line comment
# Third multi-line comment
```

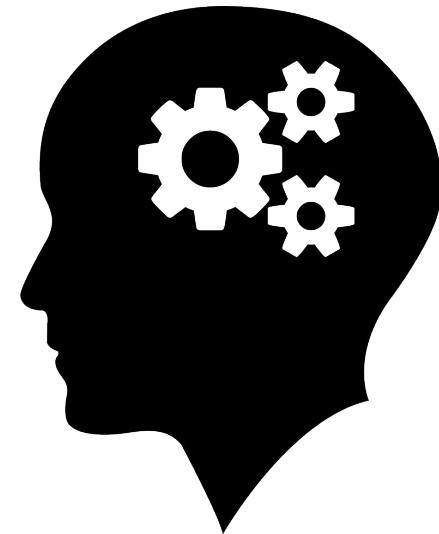
```
hello
```



Comments

Keep these in your mind ! :

- ▶ **Comments should be :**
 - ▷ **Sufficient**
 - ▷ **Necessary**
 - ▷ **Updated**



Draw or type 2 things you already know about today's topic:



Students, draw anywhere on this slide!