



Python Basics



Session-1



Python Overview Video





Before breaking the jug



Things to do...

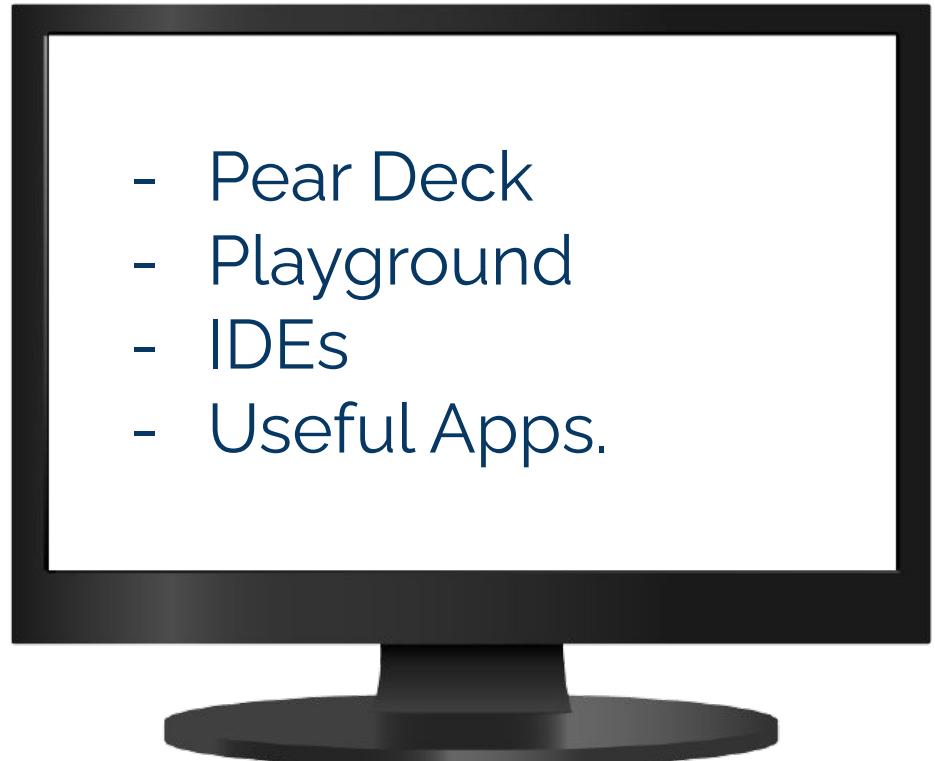


Hands off the keyboards...





Using of two screens





General Information about Python





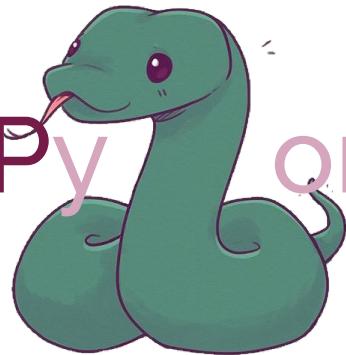
Table of Contents

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



1

What is Py on?



A snake has
never been so
cute.

What did you learn from Pre-class materials?

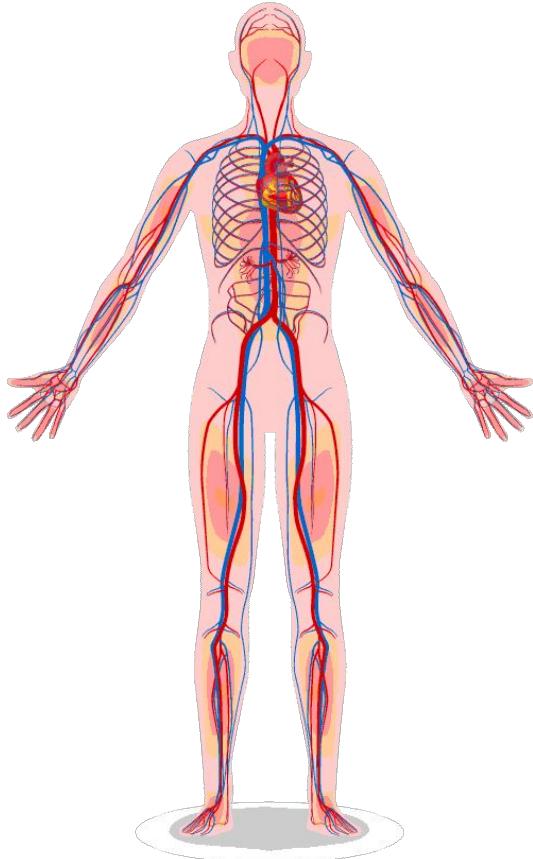
Write down three things..

- *
- *
- *



Students, write your response!

What is Python?



Mr. IT

What is Python? (review)



Less code



Pre-built libraries



Ease of learning



Platform
Independent

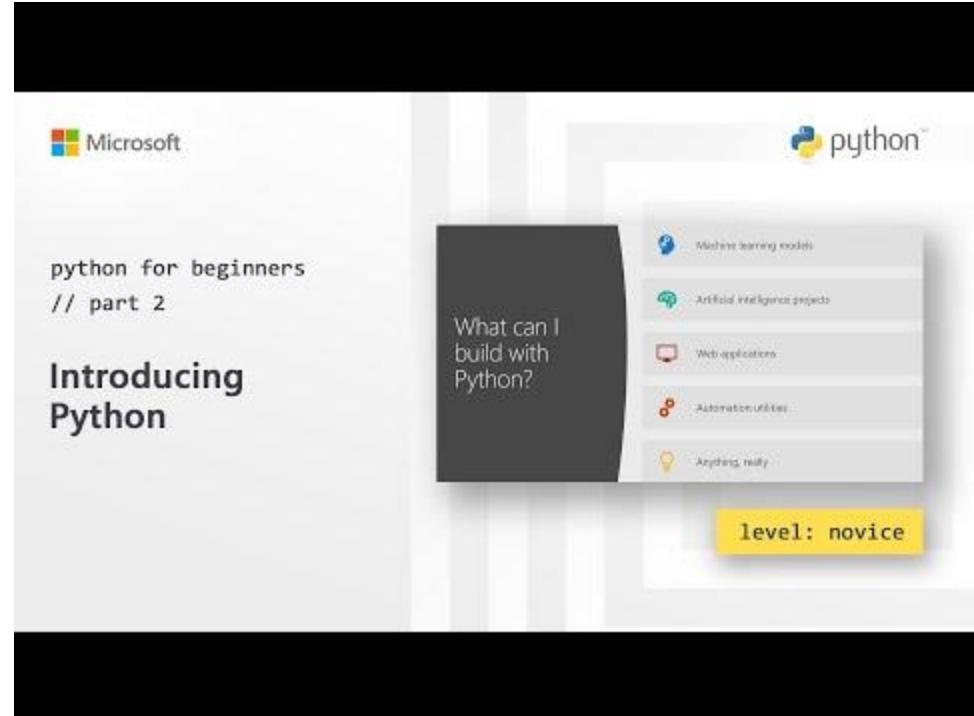


Massive Community
Support



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

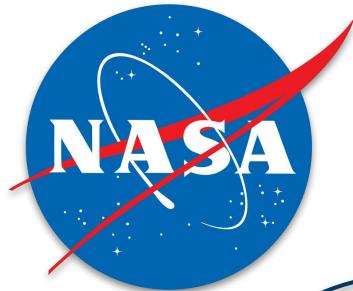


Students, write your response!

Pear Deck Interactive Slide
Do not remove this bar



What is Python?



Google

amazon.com®

N

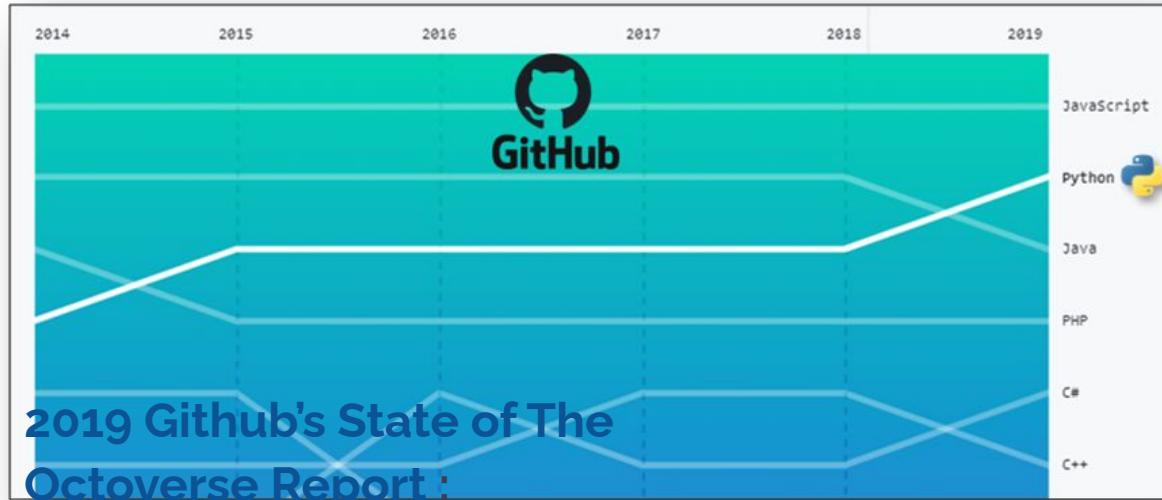
Why Python?

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.

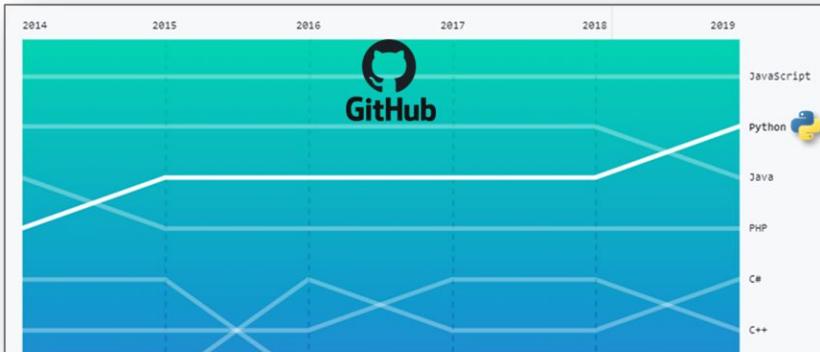


-The Second Most Popular Language



Why Python?

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



2019 Github's State of The Octoverse Report :

-The Second Most Popular Language

Oct 2019	Oct 2018	Change	Programming Language	Ratings	Change
1	1		Java	16.884%	-0.92%
2	2		C	16.180%	+0.80%
3	4	▲	Python	9.089%	+1.93%
4	3	▼	C++	6.229%	-1.36%
5	6	▲	C#	3.860%	+0.37%
6	5	▼	Visual Basic .NET	3.745%	-2.14%
7	8	▲	JavaScript	2.076%	-0.20%
8	9	▲	SQL	1.935%	-0.10%

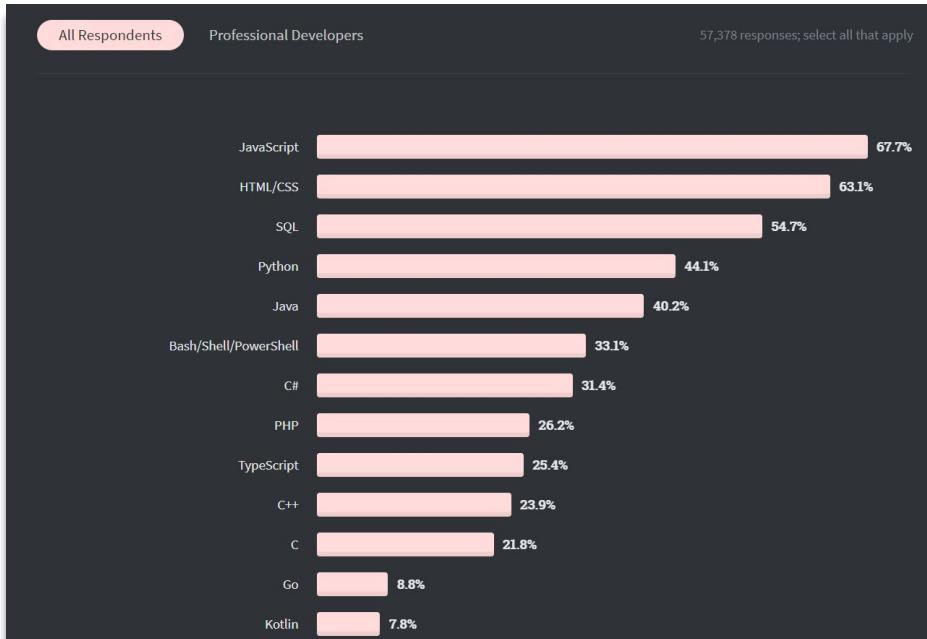
2019 Tiobes.com's Index :

-The fastest growing language



Why Python?

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



 2020 Developer Survey

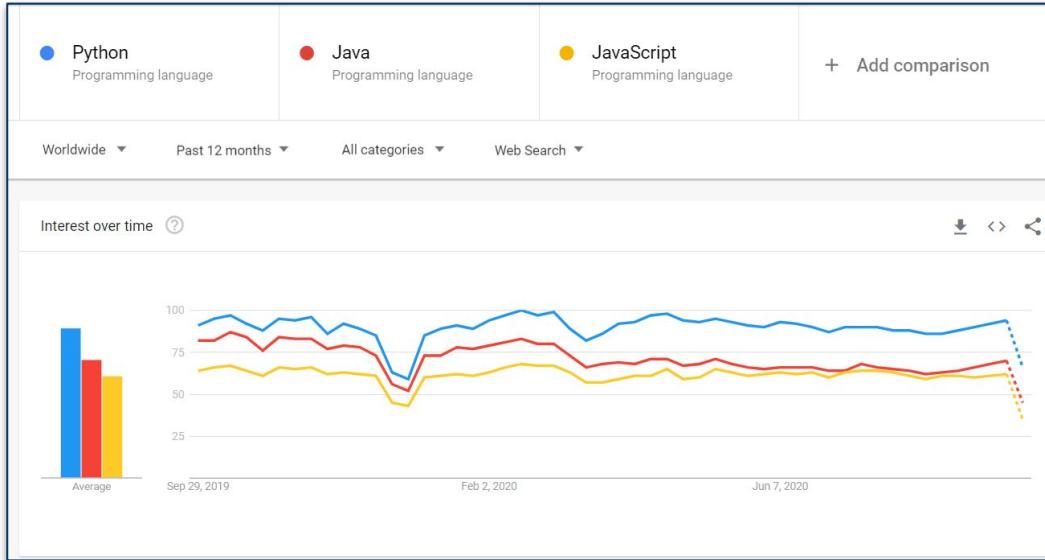
In February 2020 nearly 65,000 developers told us how they learn and level up, which tools they're using, and what they want.

The second most popular language.



Why Python?

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



2020 'trends.google.com' Report :

-The Most Popular Searched Term Among all Programming Languages



2

Historical Development of Python

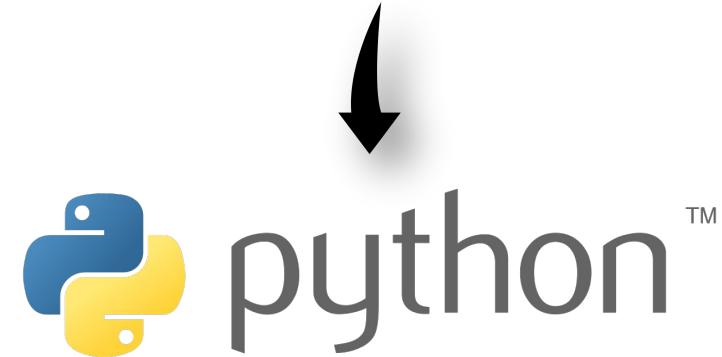
► Historical Development of Python



Guido van Rossum *(duration :4 min)*



The logo





3

Review of Tools & Installations

Review of Tools & Installations



How and where to run your Python codes?

Python Playground

Hi Joseph, you can write your code using the editor below. Once you write the code, click the run button to see the result.

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

Run

Output

```
hello world
```

Untitled

File Edit View Insert Runtime Tools Help All c...

Comment Share ⚙️ J

+ Code + Text

RAM Disk ✎ Editing

Google Colab Jupyter Notebook

```
[1] print ("hello")
hello
[ ]
```

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

True

False



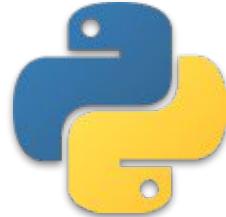
Pear Deck



Students choose an option



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'

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)
```



SWAY[©]

Students, write your response!

Pear Deck Interactive Slide
Do not remove this bar

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



PEP 8 Conventions

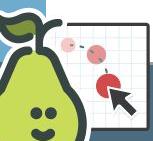




Table of Contents

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

How was pre-class content ?



Students, drag the icon!

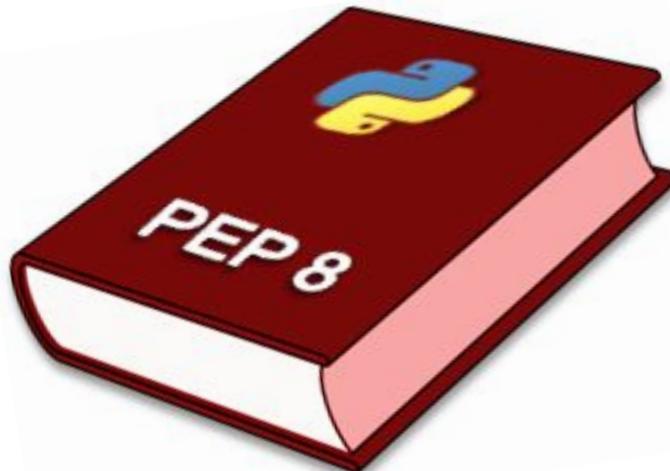


What is PEP 8?

PEP 8



Python Enhancement Proposal

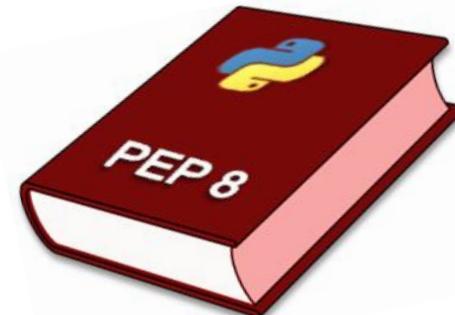




What is PEP 8?

PEP 8 is a style guide about consistency.

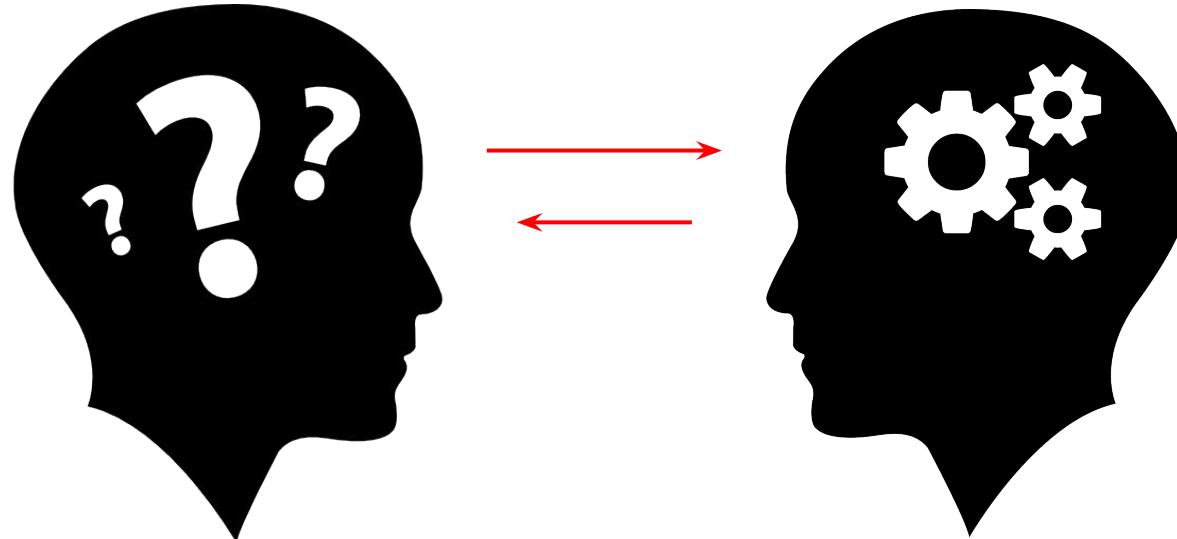
- ▶ Consistency with this style guide is **important**.
- ▶ Consistency within a project is **more important**.
- ▶ Consistency within one module or function is the **most important coding aspect**.





What is PEP 8?

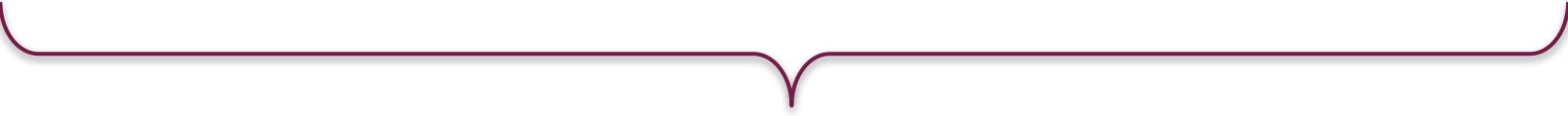
COMMUNICATE with each other!



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

- 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;` **NO** : `if y == 3 : print x , y ;`

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



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**

