PYTHON FUNDAMENTALS

Properties and methods

LEARNING OBJECTIVES

- To understand properties and methods
- To understand different data types
- To understand libraries
- To create simple programs on Visual Studio Code

PYTHON

Readable and Maintainable Code

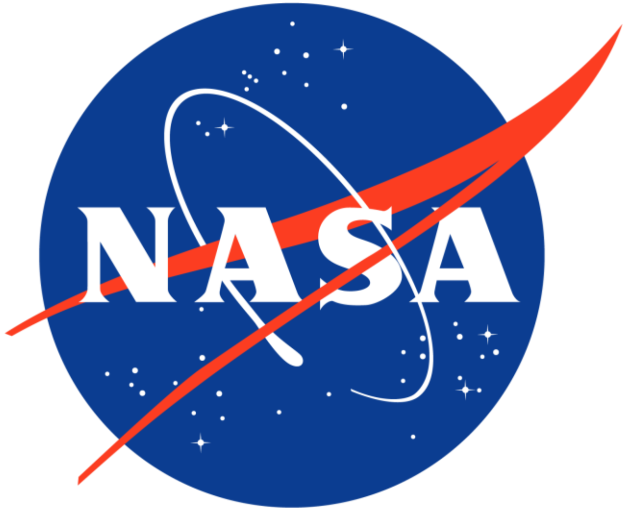
Quick to start

Loads of beginner resources

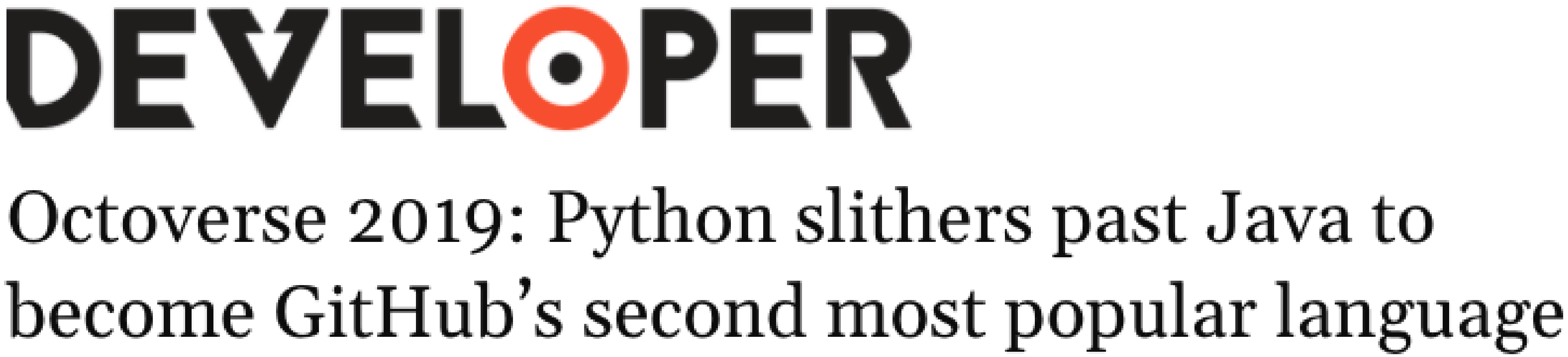
NETFLIX

Google

YouTube


NASA

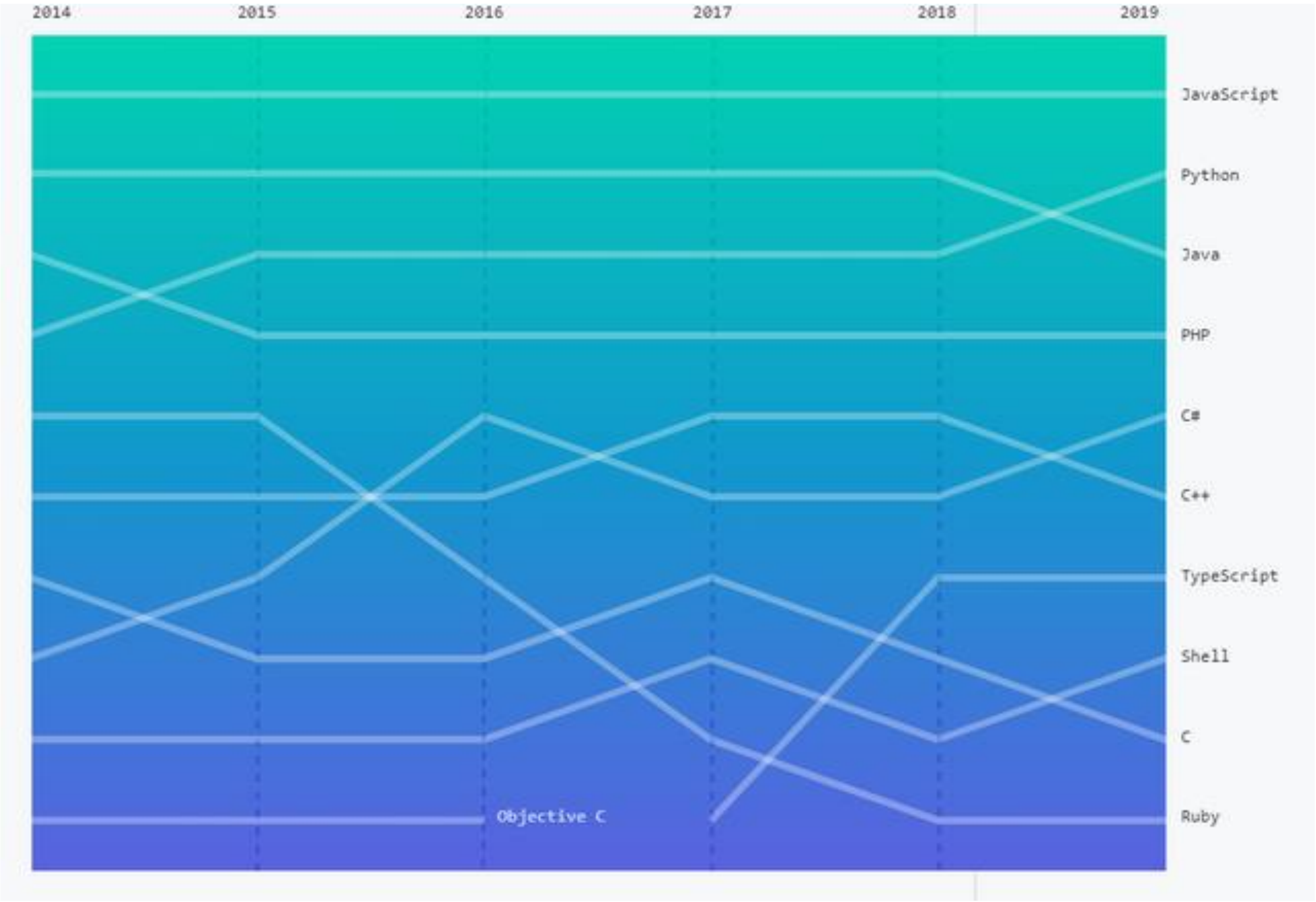
It’s also one of the most popular language in the world



DEVELOPER

Octoverse 2019: Python slithers past Java to become GitHub’s second most popular language

Top 10 most popular languages of 2019 according to GitHub



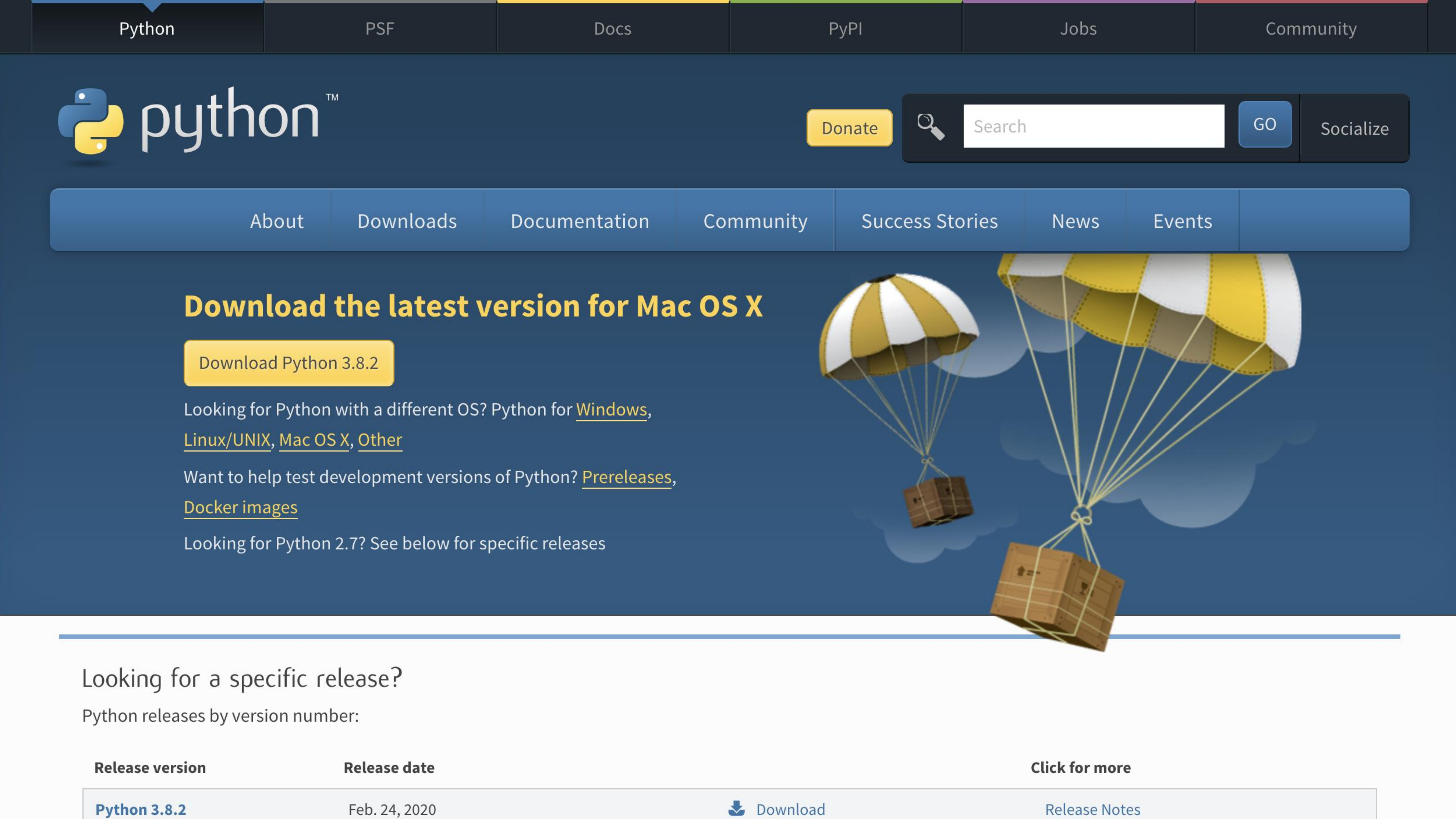
Too long; didn’t read?

It’s a very good language to learn!

Printing information

um... to a printer?

TO VSCODE



Download the latest version for Mac OS X

Looking for Python with a different OS? Python for Windows, Linux/UNIX, Mac OS X, Other

Want to help test development versions of Python? Prereleases, Docker images

Looking for Python 2.7? See below for specific releases

Looking for a specific release?

Python releases by version number:

Release version	Release date		Click for more
Python 3.8.2	Feb. 24, 2020	Download	Release Notes

A LITTLE TIP

WINDOWS

Copy anything

Use the keys ctrl, c

Paste

Use the keys ctrl, v

Cut

Use the keys ctrl, x

Undo

Use the keys ctrl, z

Save your work

Use the keys ctrl, s

MAC

Copy anything

- Use the keys cmd, c

Paste

- Use the keys cmd, v

Cut

- Use the keys cmd, x

Undo

- Use the keys cmd, z

Save your work

- Use the keys cmd, s

Print function

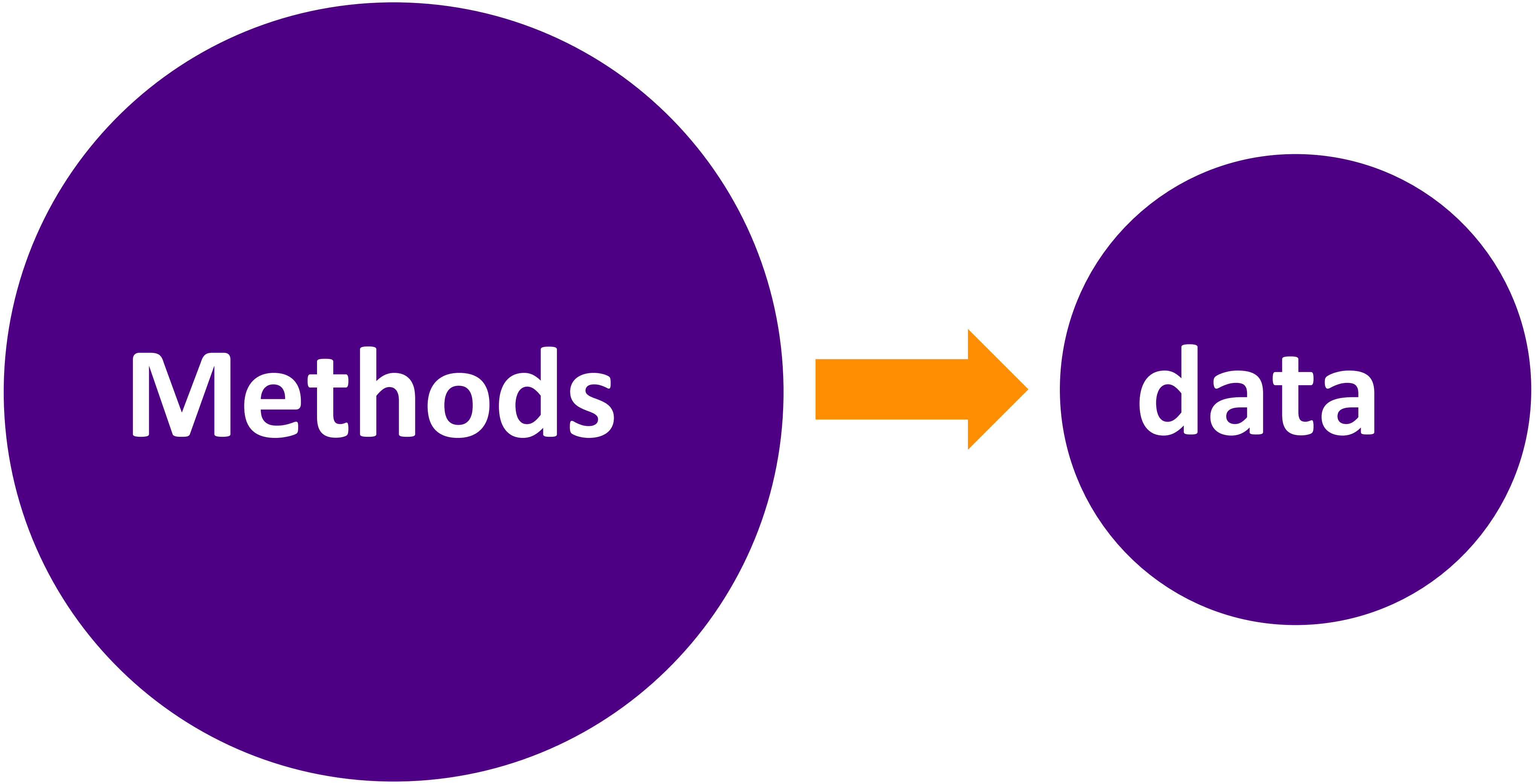
```
print("hello world")
```

Commenting

```
#this is a comment
```

Everyone loves a bit of data

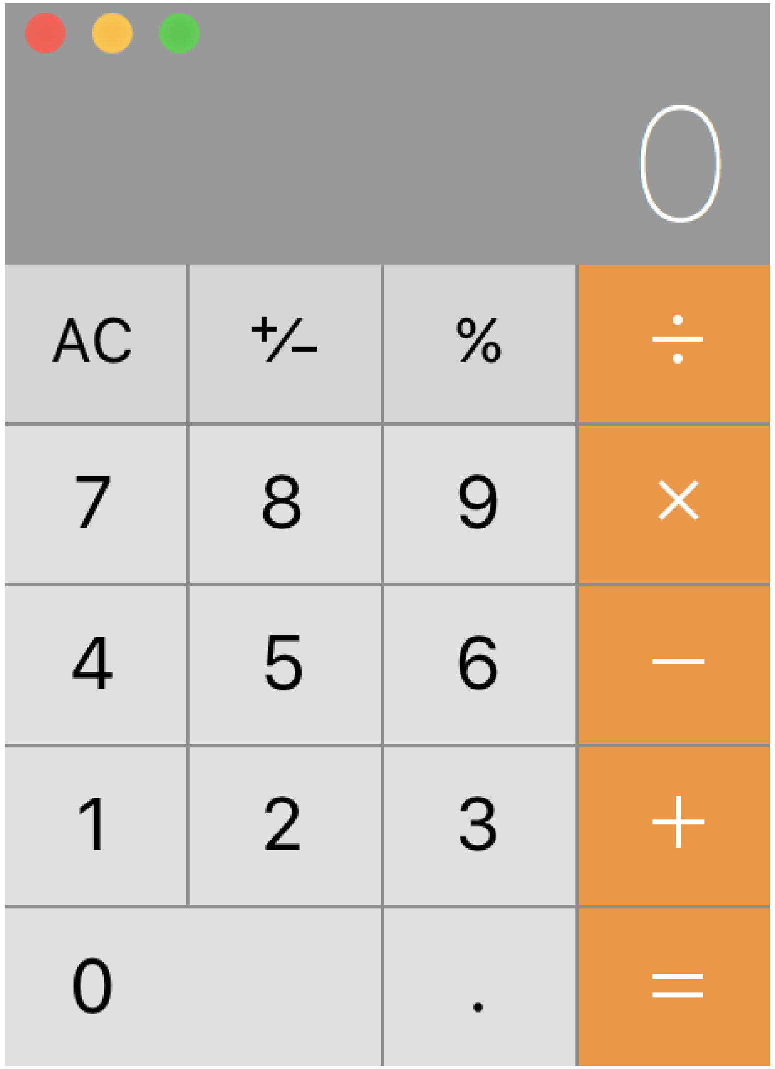
If we break coding down to its simplest and snappiest



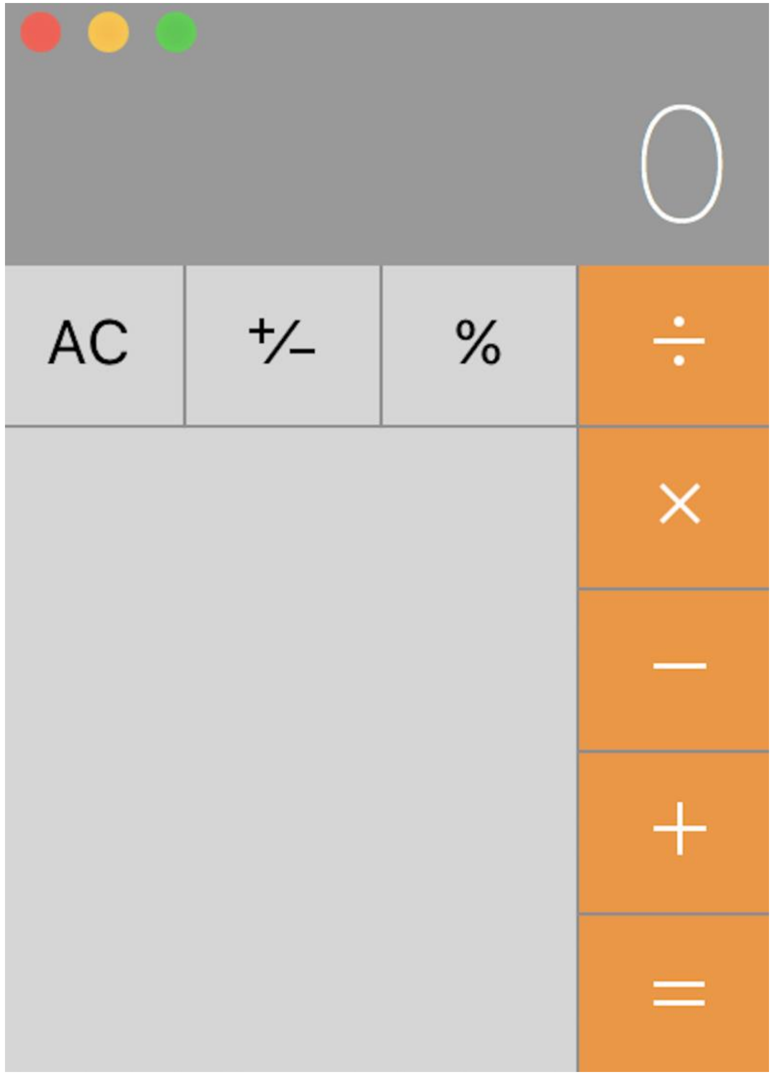
Methods

data

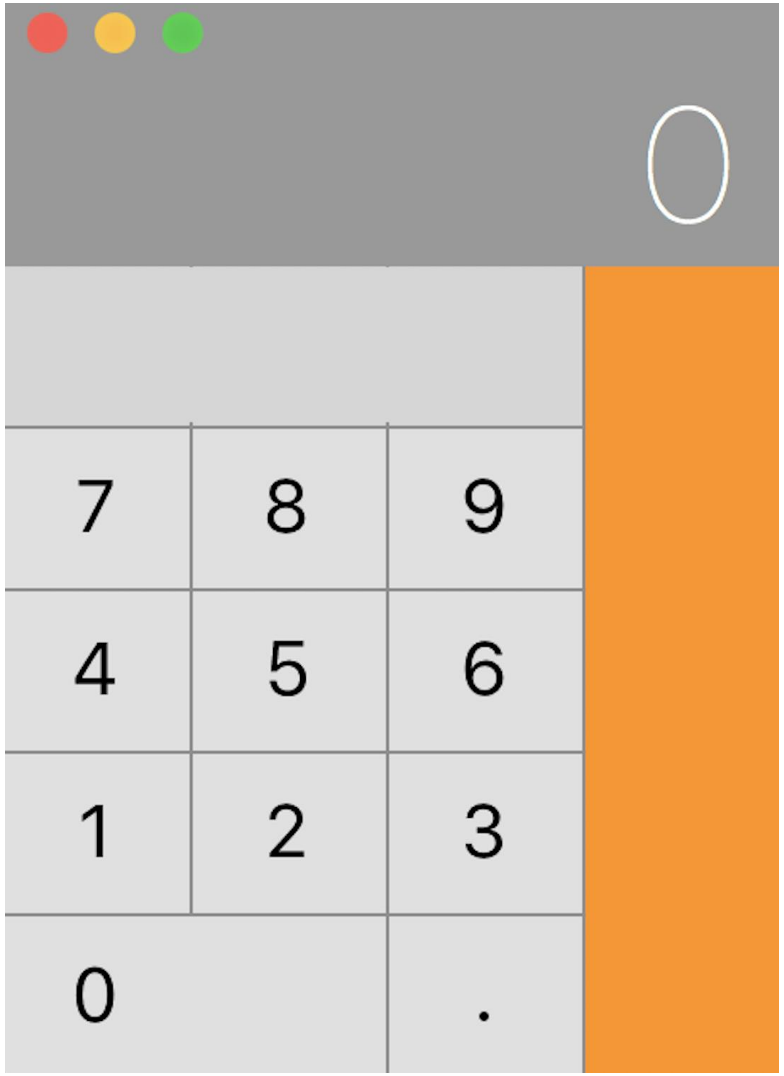
Everyone loves a calculator



But how much would you love it if you took this to your maths exam?



Perhaps the cruelest of them all!



Methods and data intimately linked

Working with data types

Data types refers to the kind of data that we're asking the computer to work with

Working with data types

In Python, we have a few simple kinds of data

These are:

String : for representing text

Integer: for representing whole number

Floating point: for decimals

None : for nothing

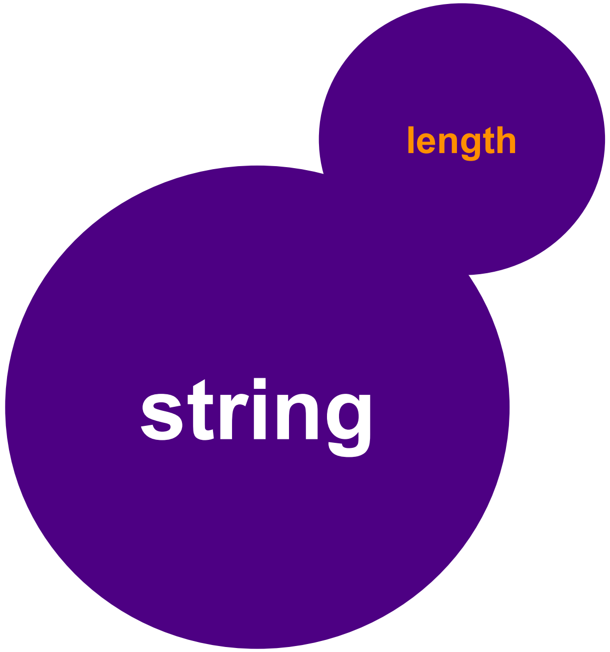
Boolean: for True and False

Get on my property

```python
print("What data type am I?")
```

Good stuff,
it is indeed a string

Properties



length

string

Strings have associated data or additional information available

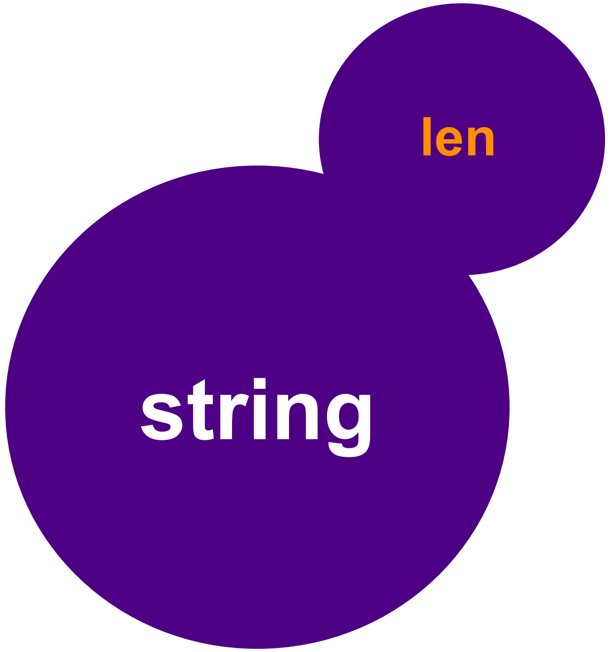
```
print("hello")
```

How many letters does hello have?

To VSCode

```
print(len("hello"))
```

***finding the length property of this string**



len

string

Strings have associated data or linked data

```
print(len("hello"))
```

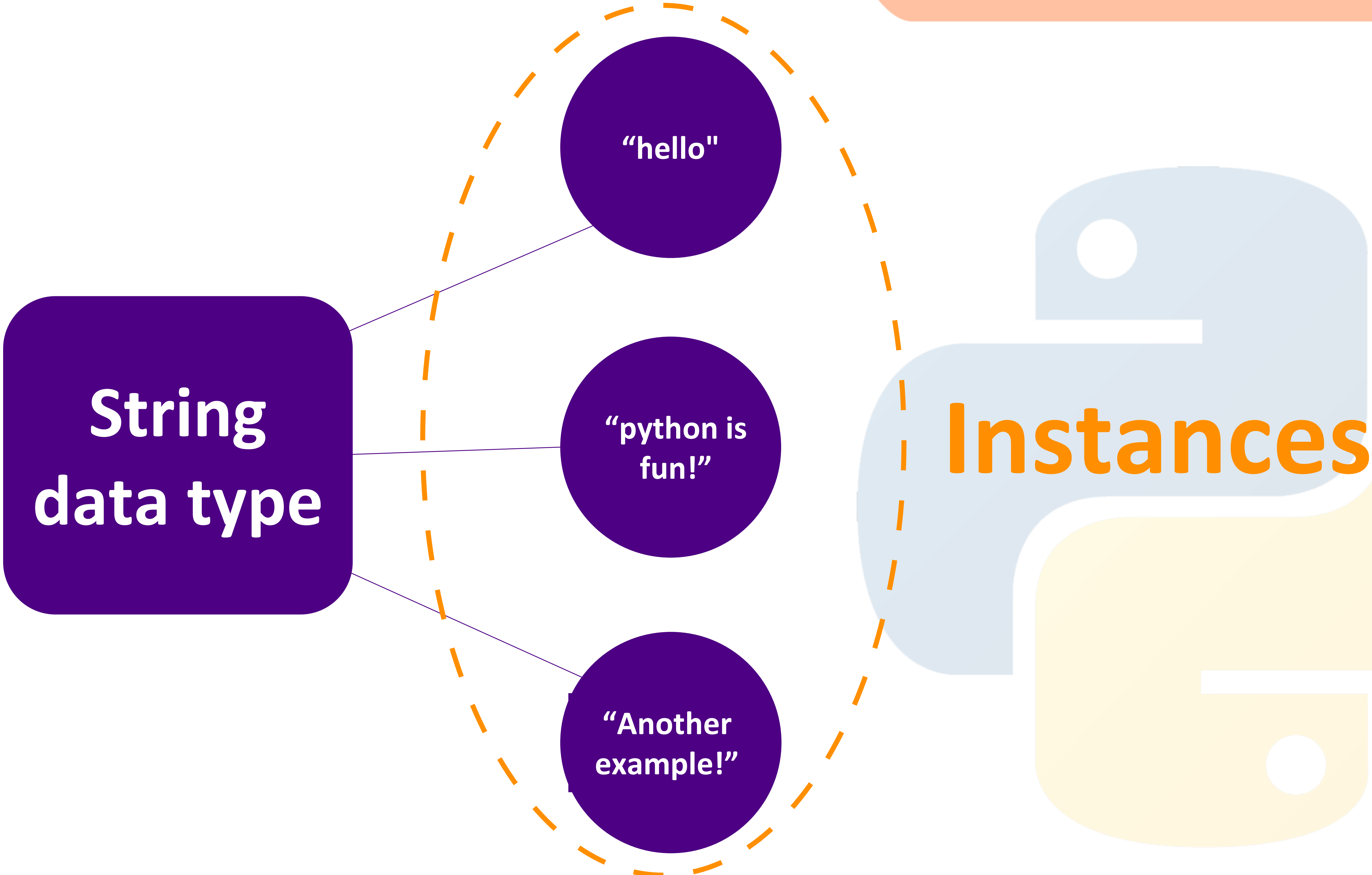
Finding a particular character in a string

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

Finding a particular character in a string

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

*finding the first character of this string, note that index begins at 0



String data type

“hello"

“python is fun!”

“Another example!”

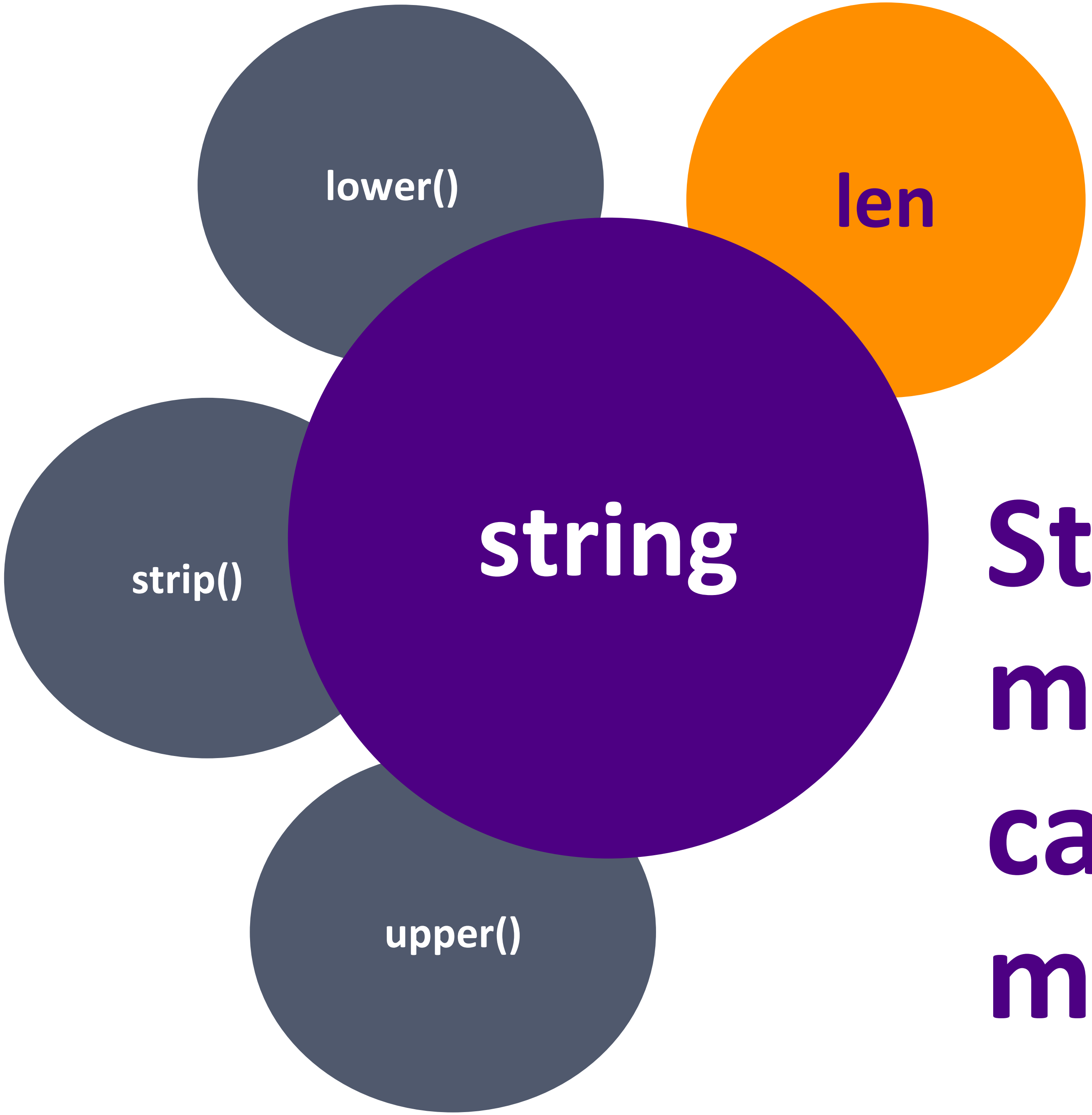
Instances

These instances live in the computer’s memory (Random Access Memory)

Methods in the madness

Methods and data intimately linked

The built-in data types have built-in associated methods



Strings have methods that we can use to manipulate them

Methods let us do stuff!

Unlike properties, which are essentially just information

To VSCode

```
print("hello".upper())
```

Dot notation

```
print("hello".upper())
```

Dot notation

```
"hello".upper()
object.method
```

Methods

There are other methods you will be able to explore on what they do:

- upper()
- lower()
- capitalize()
- count()
- find()
- replace()
- strip()

Ssssh.

Libraries.

In coding, libraries give us access to a bunch of features that we don’t have to code ourselves

So far, we’ve stuck to `print` and… that’s about it

`print` is in built-in library

Let’s see the power of a library in action.
A classic example : generating a random number

There are many libraries out there we can use

```
import random
```

random is a library in Python, this is not a built-in library. If we want to use a library we have to import it first.

```
import random

print(    )
```

Parameters

```
import random

print(random.random())
```

Give that a go. What happens?

```
import random

print(random.random())
```

Dot notation

Generates a random number between 0 and 1, including 0 only.

Generating a random number(1)

- random is a library in python
- random is a method in the random library
- random doesn’t need any parameter

```
random.random()
```

```
import random

print(random.uniform(1, 10))
```

Give that a go. What happens?

```
import random

print(random.uniform(1,10))
```

Dot notation

Parameters

Generates a random number between 1 and 10, inclusive.

Generating a random number(2)

- random is a library in python
- uniform is a method in the random library
- uniform takes two parameters:
  - an lower, and
  - a upper bound

```
random.uniform(lower,upper)
```

```
import random

print(random.randint(1,10))
```

Give that a go. What happens?

```
import random

print(random.randint(1,10))
```

Dot notation

Parameters

Generates a random integer between 1 and 10, inclusive.

Generating a random number(3)

- random is a library in python
- randint is a method in the random library
- randint takes two parameters:
  - an lower, and
  - a upper bound

```
random.randint(lower,upper)
```

To summarise

To summarise

- random is a library in python
- random, uniform, randint are methods in the random library
  - random.random()
  - random.uniform(1,10)
  - random.randint(1,10)
- They may or may need parameter(s)

To summarise

```
import random


print(random.random())
#Generates a random number between 0 and 1, including 0 only.
print(random.uniform(1, 10))
#Generates a random number between 1 and 10, inclusive.
print(random.randint(1,10))
#Generates a random integer between 1 and 10, inclusive.
```

To summarise

Import all necessary libraries at the start of the program.

```
import random

print(random.random())
#Generates a random number between 0 and 1, including 0 only.
print(random.uniform(1, 10))
#Generates a random number between 1 and 10, inclusive.
print(random.randint(1,10))
#Generates a random integer between 1 and 10, inclusive.
```

To summarise

Then the main code

```
import random


print(random.random())

#Generates a random number between 0 and 1, including 0 only.

print(random.uniform(1, 10))

#Generates a random number between 1 and 10, inclusive.

print(random.randint(1,10))

#Generates a random integer between 1 and 10, inclusive.
```

from library import method

*extension

To the next level…

```
import random
print(random.random())
print(random.uniform(1, 10))
print(random.randint(1,10))
from random import random, randint, uniform
print(random())
print(uniform(1, 10))
print(randint(1,10))
```

Both are the same!

*extension

To the next level…

```
import random
print(random.random())
print(random.uniform(1, 10))
print(random.randint(1,10))
from random import random, randint, uniform
print(random())
print(uniform(1, 10))
print(randint(1,10))
```

Both are the same!

*extension

To the next level…

```
import random
print(random.random())
print(random.uniform(1, 10))
print(random.randint(1,10))
from random import random, randint, uniform
print(random())
print(uniform(1, 10))
print(randint(1,10))
```

Both are the same!

*extension

To the next level…

```
import random
print(random.random())
print(random.uniform(1, 10))
print(random.randint(1,10))
from random import random, randint, uniform
print(random())
print(uniform(1, 10))
print(randint(1,10))
```

Both are the same!

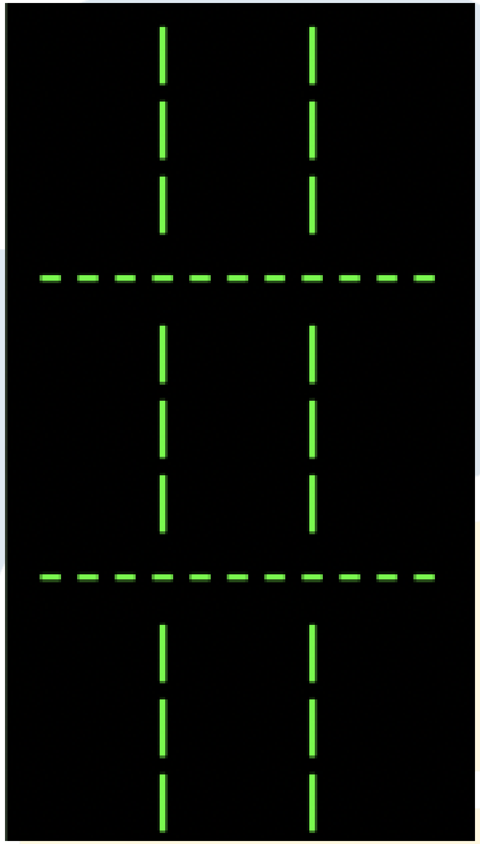
*extension

LEARNING OBJECTIVES

- To understand properties and methods
- To understand different data types
- To understand libraries
- To create simple programs on Visual Studio Code

Activity(1)

Have a go at printing a grid like this to the terminal



```
   |   |
   |   |
   |   |
-----------
   |   |
   |   |
   |   |
-----------
   |   |
   |   |
   |   |
```

*Not using any special method or different way other than print

Activity(2): Methods

Look into these methods write an example of each with an explanation of how each one works:

- upper()
- lower()
- capitalize()
- count()
- find()
- replace()
- strip()

Activity(3): Methods

Are there other useful methods…

- from built-in library?
- from random library?
  - https://docs.python.org/3/library/random.html