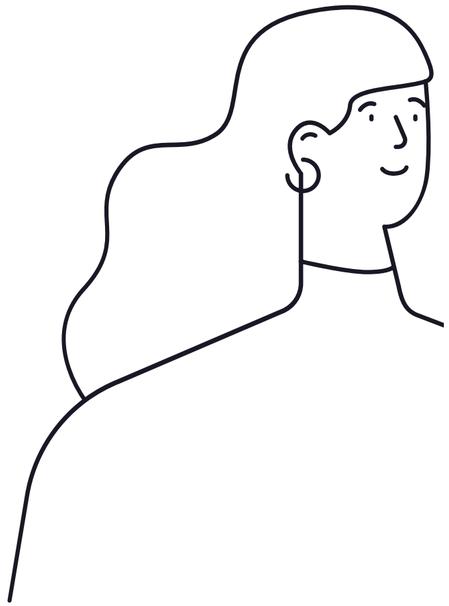
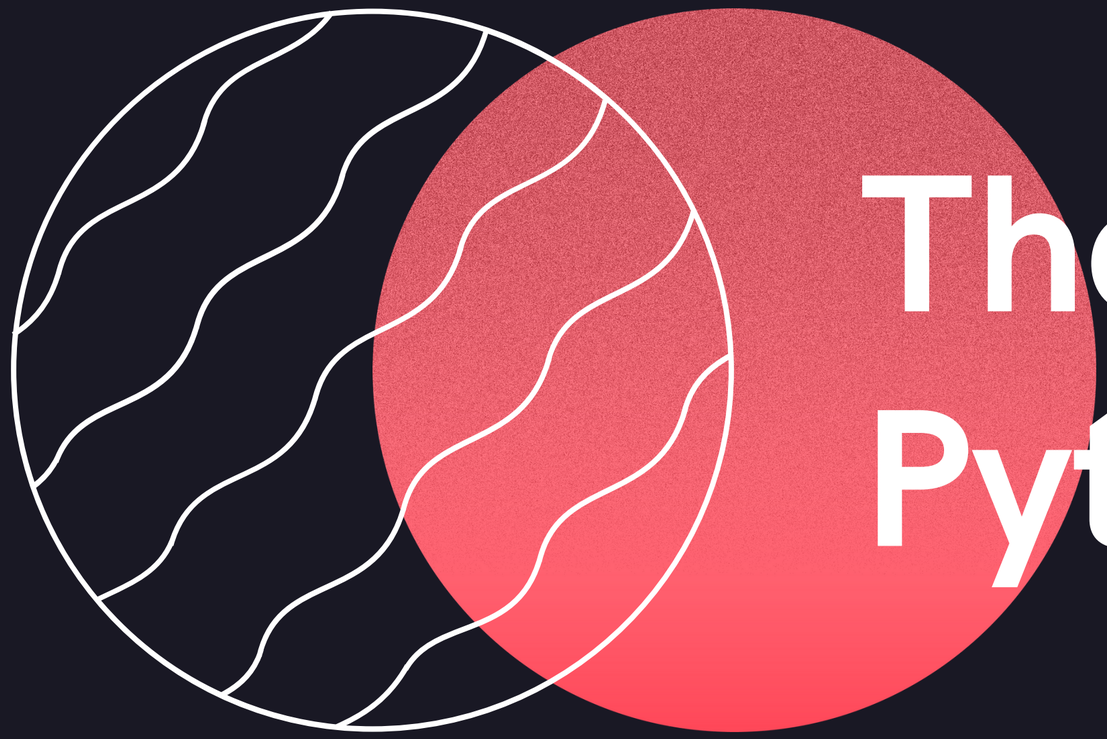
Welcome Everyone!

We will wait for others to join in!

KNOW ABOUT ME:

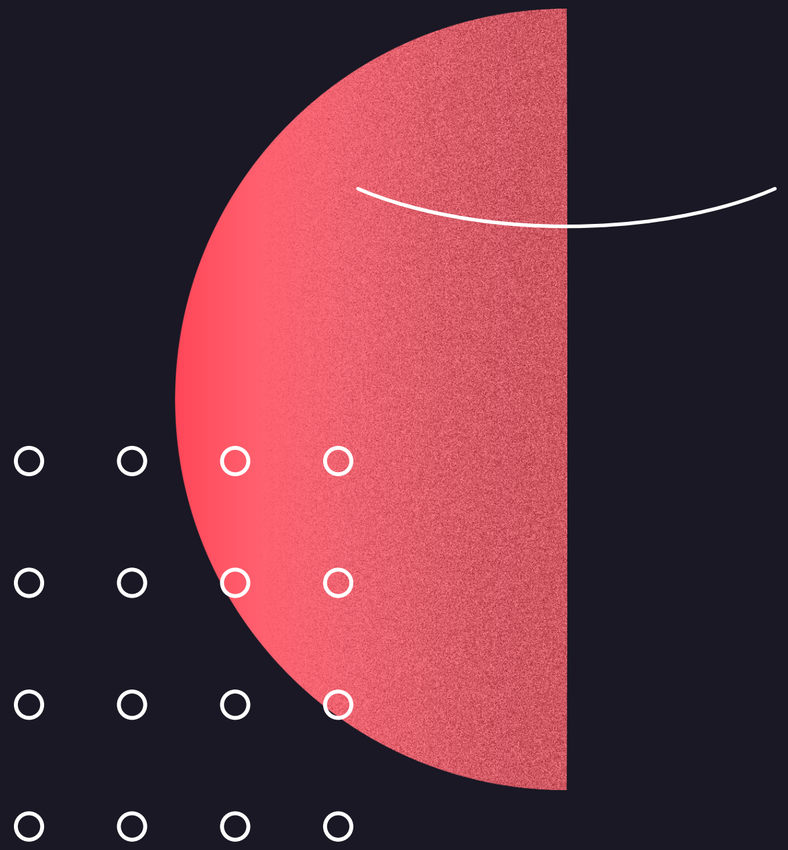


We Will start in 10



The Basics of Python

With, Aaryan Kapur



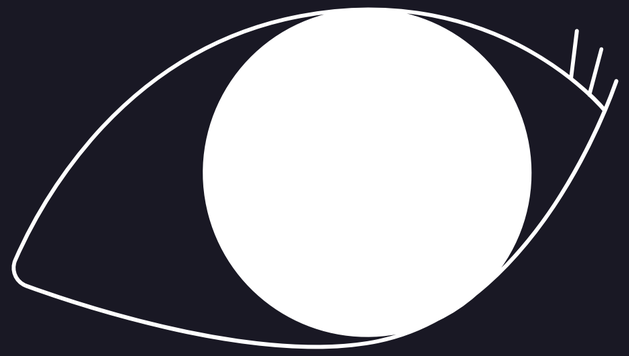
Let's begin with some short introductions!

SOME APPLICATIONS OF PYTHON

Created way back in 1989 by Guido Van Rossum, Python has some real impact on the world as we know it!

Web Development	Desktop GUI	Image Processing	Business Applications
Game Development	Image Processing	Text Processing	Education Programs
Computer Vision	IOT	Prediction	Processing
Artificial Intelligence and Machine Learning	Audio and Video Preprocessing and Editing	Data Science, Data Analysis, Data Visualization	Scientific and Numeric Applications

LET'S INSTALL PYTHON!



SCREEN SHARE!

DATA TYPES IN PYTHON

Text Type: str
Numeric Types: int, float, complex
Sequence Types: list, tuple, range
Mapping Type: dict
Set Types: set, frozenset
Boolean Type: bool
Binary Types: bytes, bytearray

```
x = "Hello World"                          str
x = 20                                     int
x = 20.5                                   float
x = 1j                                     complex
x = ["apple", "banana", "cherry"]          list
x = ("apple", "banana", "cherry")          tuple
x = range(6)                               range
x = {"name" : "John", "age" : 36}          dict
x = {"apple", "banana", "cherry"}          set
x = frozenset({"apple", "banana"})         frozenset
x = True                                   bool
```

PLAYING WITH STRINGS

```
b = "Hello, World!"
print(b[0:5])
```

→ Hello

```
a = "Hello"
b = "World"
c = a + b
print(c)
```

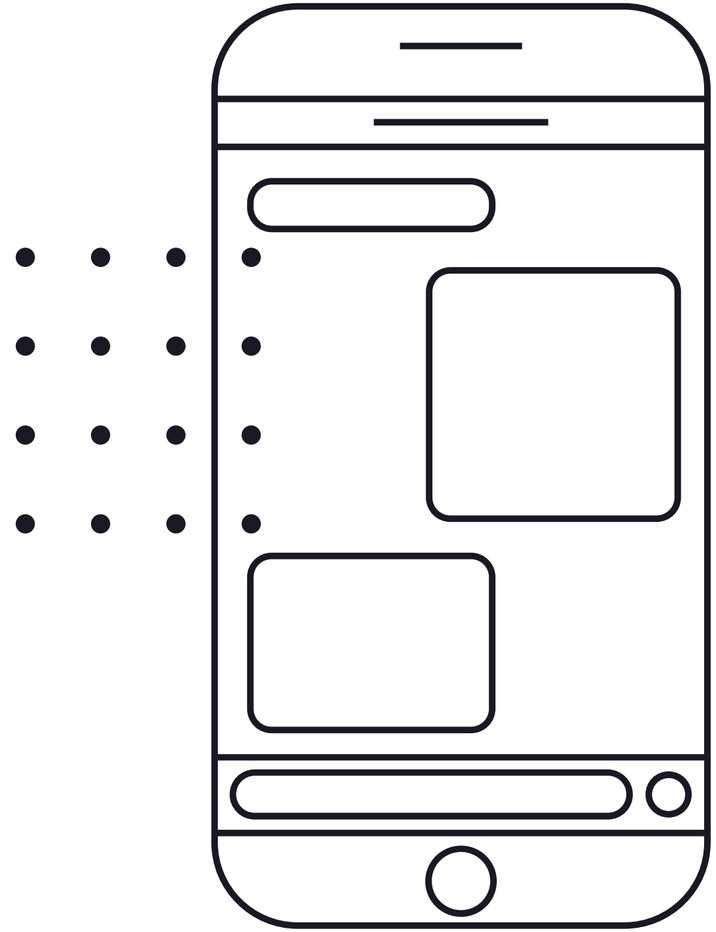
→ HelloWorld

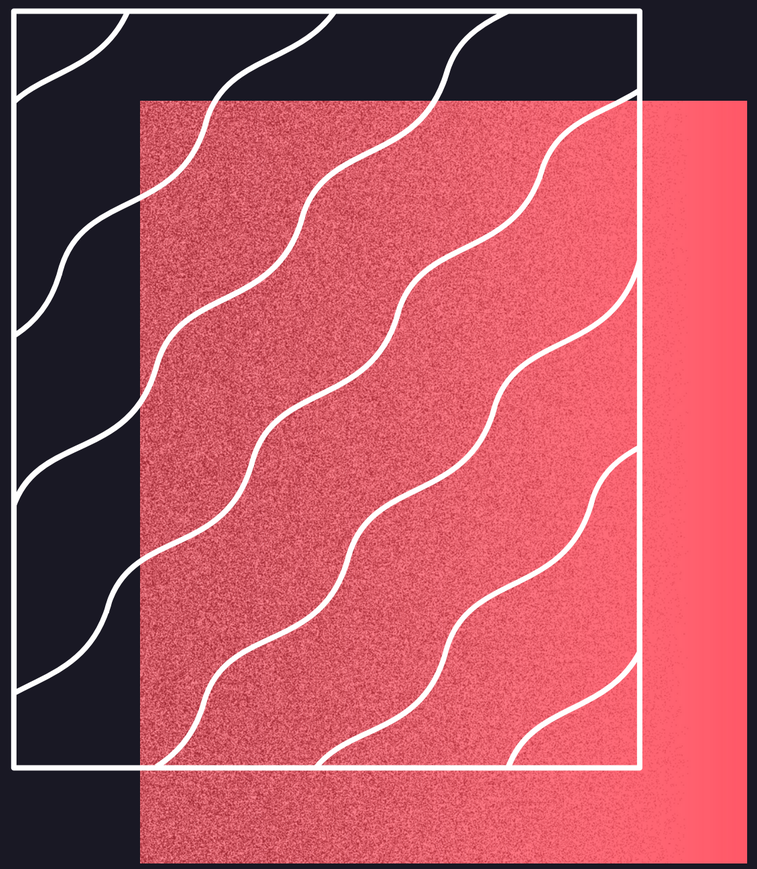
```
print("My
name is
John, I am "
+ 90)
```

→ My name is John,
I am 90

```
print("{} is more
than {}"
.format(30, 20))
```

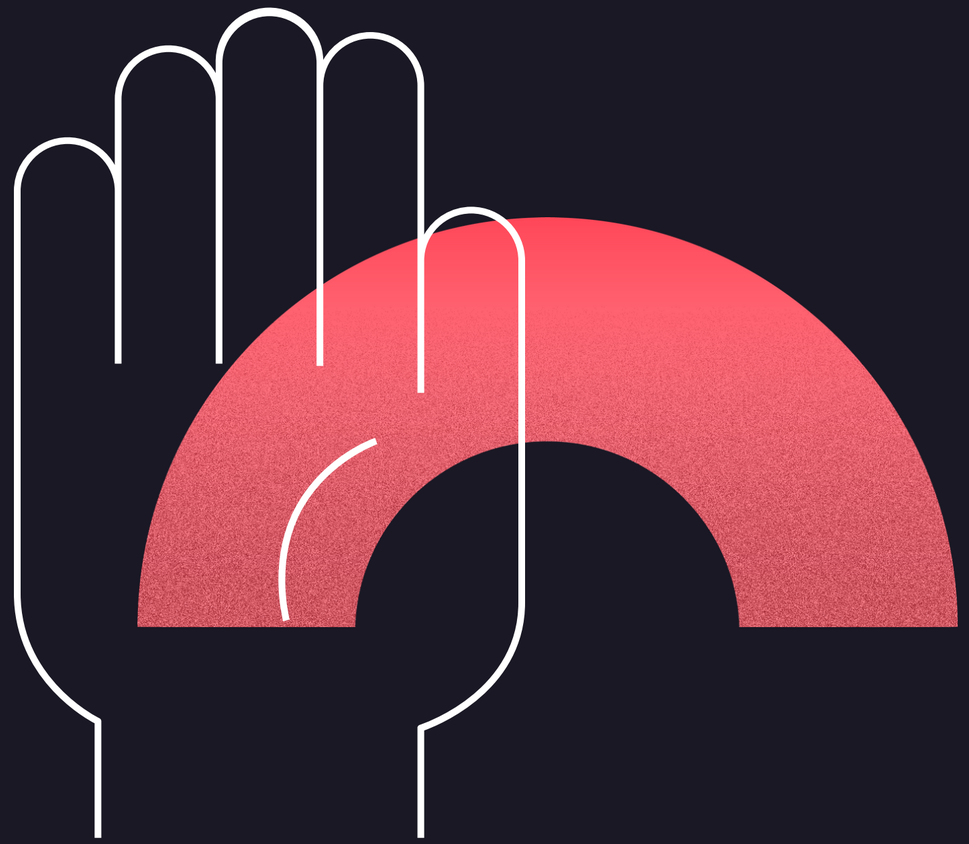
→ 30 is more than 20





MATHEMATIC OPERATORS

+	Addition	x + y
-	Subtraction	x - y
*	Multiplication	x * y
/	Division	x / y
%	Modulus	x % y
**	Exponentiation	x ** y
//	Floor division	x // y

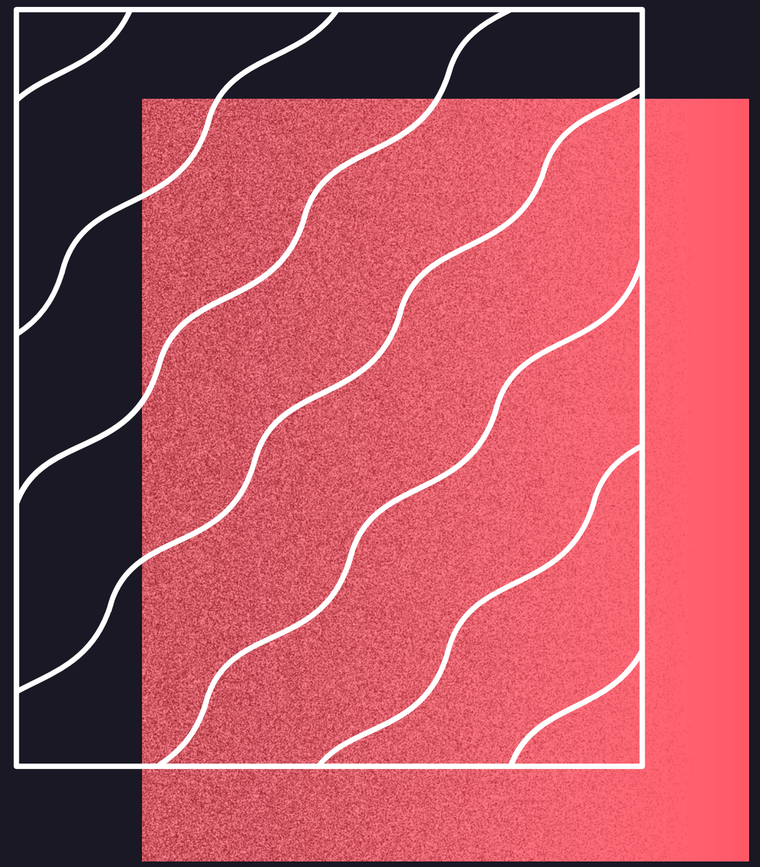


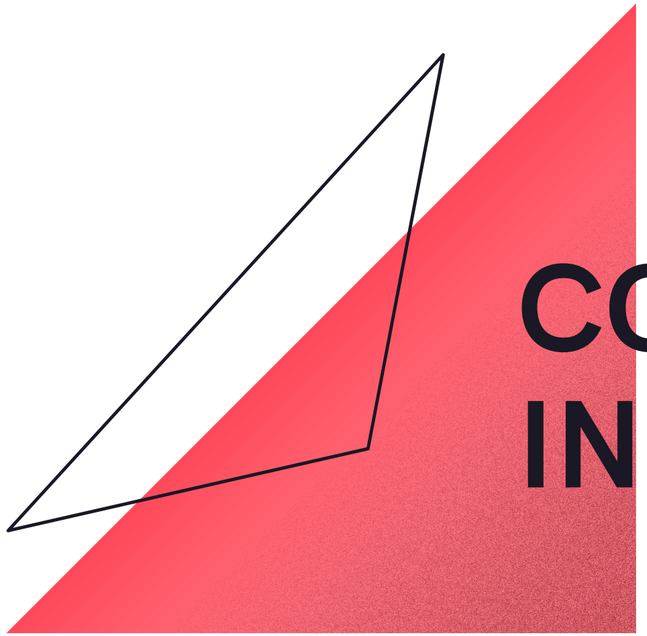
ASSIGNMENT OPERATORS

=	x = 5	x = 5
+=	x += 3	x = x + 3
-=	x -= 3	x = x - 3
*=	x *= 3	x = x * 3
/=	x /= 3	x = x / 3
%=	x %= 3	x = x % 3
//=	x //= 3	x = x // 3
**=	x **= 3	x = x ** 3
&=	x &= 3	x = x & 3
|=	x |= 3	x = x | 3
^=	x ^= 3	x = x ^ 3
>>=	x >>= 3	x = x >> 3
<<=	x <<= 3	x = x << 3

COMPARISON OPERATORS

==	Equal	x == y
!=	Not equal	x != y
>	Greater than	x > y
<	Less than	x < y
>=	Greater than or equal to	x >= y
<=	Less than or equal to	x <= y





COLLECTIONS IN PYTHON

List is a collection that is ordered and changeable. Allows duplicate members.

Tuple is a collection that is ordered and unchangeable. Allows duplicate members.

Set is a collection that is unordered and unindexed. No duplicate members.

Dictionary is a collection that is ordered* and changeable. No duplicate members.

LISTS IN PYTHON

```
list= ["abc", 34, True, 40, "male"]
```

TUPLES IN PYTHON

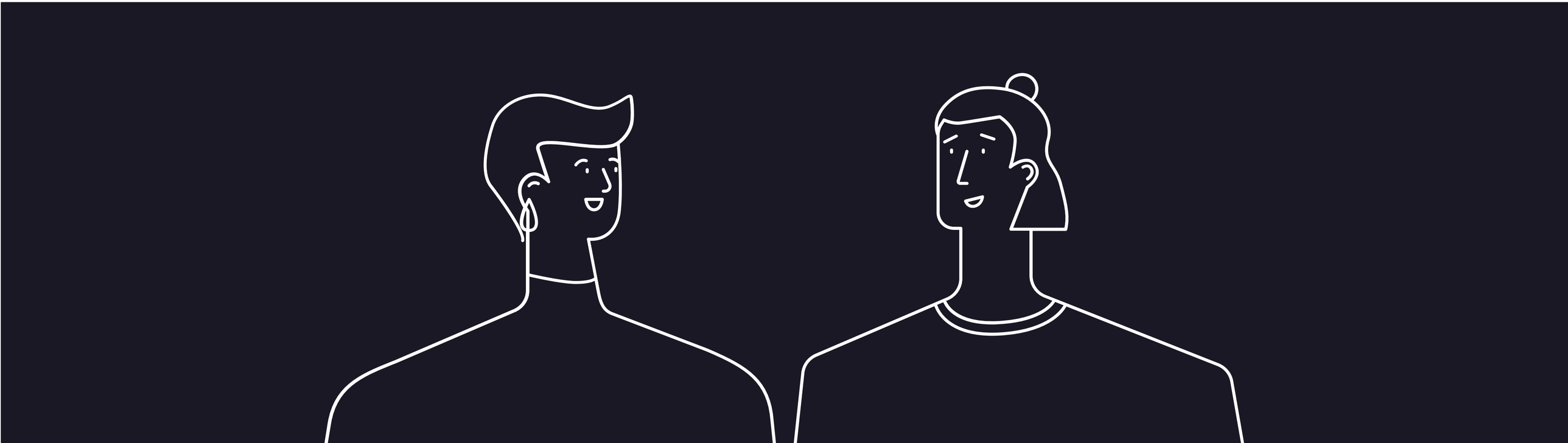
```
tuple = ("apple", "banana", "cherry")
```

SETS IN PYTHON

```
set= {"apple", "banana", "cherry"}
```

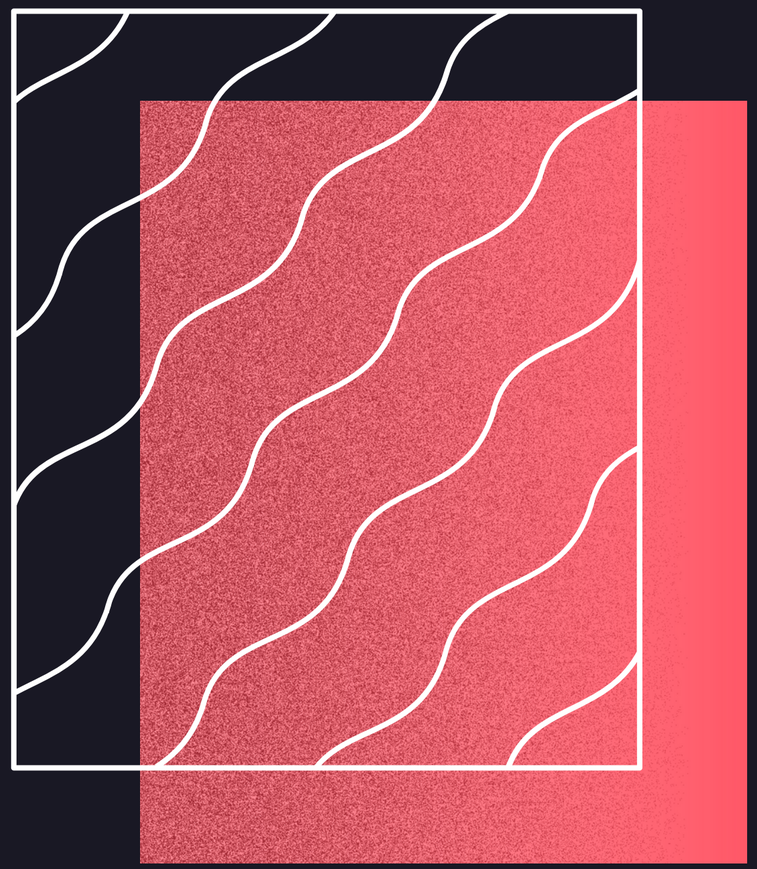
DICTIONARY IN PYTHON

```
dict = {"brand": "Ford",
"model": "Mustang", "year": 1964}
```



DIFFERENCE BETWEEN DATATYPES

List	Tuple	Set	Dictionary
List is a non-homogeneous data structure which stores the elements in single row and multiple rows and columns	Tuple is also a non-homogeneous data structure which stores single row and multiple rows and columns	Set data structure is also non-homogeneous data structure but stores in single row	Dictionary is also a non-homogeneous data structure which stores key value pairs
List can be represented by [ ]	Tuple can be represented by ( )	Set can be represented by { }	Dictionary can be represented by { }
List allows duplicate elements	Tuple allows duplicate elements	Set will not allow duplicate elements	Set will not allow duplicate elements but keys are not duplicated
List can use nested among all	Tuple can use nested among all	Set can use nested among all	Dictonary can use nested among all
Example: [1, 2, 3, 4, 5]	Example: (1, 2, 3, 4, 5)	Example: {1, 2, 3, 4, 5}	Example: {1, 2, 3, 4, 5}
List can be created using **list()** function	Tuple can be created using **tuple()** function.	Set can be created using **set()** function	Dictonary can be created using **dict()** function.
List is mutable i.e we can make any changes in list.	Tuple is immutable i.e we can not make any changes in tuple	Set is mutable i.e we can make any changes in set. But elements are not duplicated.	Dictionary is mutable. But Keys are not duplicated.
List is ordered	Tuple is ordered	Set is unordered	Dictionary is ordered
Creating an empty list l=[]	Creating an empty Tuple t=()	Creating a set a=set() b=set(a)	Creating an empty dictionary d={}



CONDITIONALS STATEMENTS IN PYTHON

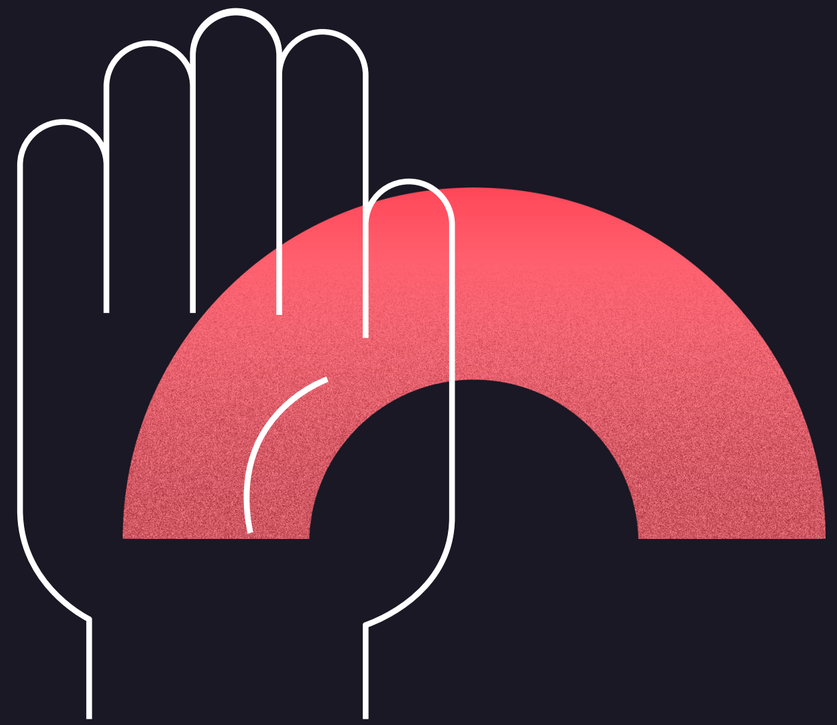
```
a = 10
b = 20

if(a>b):
    print("a is the largest")
elif(b>a):
    print("b is the largest")
else:
    print("a is equal to b")
```

LOOPS IN PYTHON

While Loop — As long as a condition is true

```
i = 1
while i < 6:
  print(i)
  i += 1
```

```
1
2
3
4
5
```



For Loop — Once for each item in a list, tuple, set, range, etc..

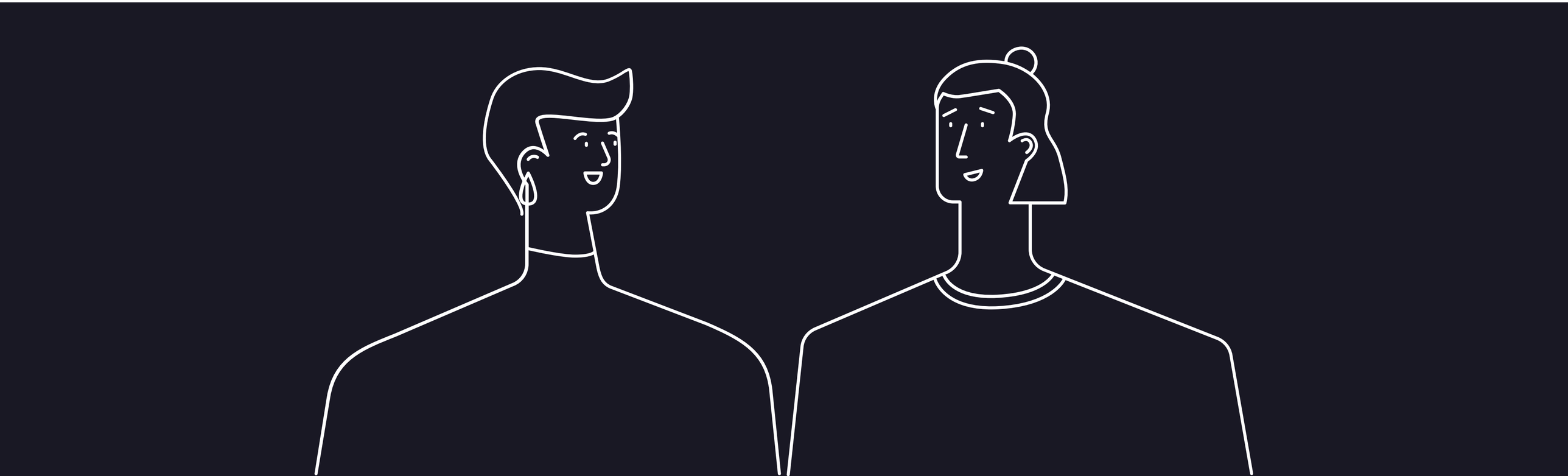
```
for x in range(1,6):
print(x)
```

```
1
2
3
4
5
```

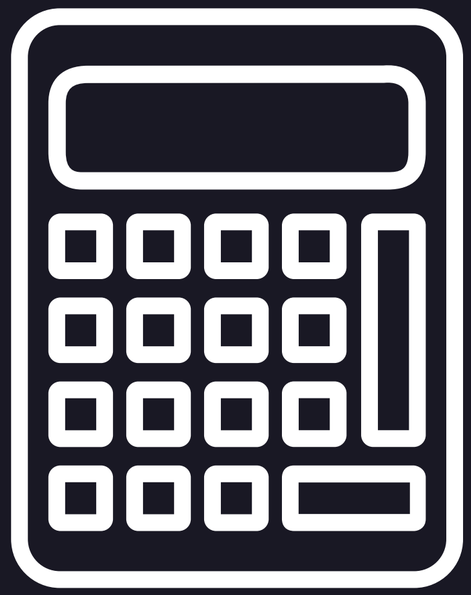
COMING UP NEXT:

- A deeper look into what we covered just now.
- A deeper understanding of loops and it's applications

- Functions, Classes, Objects, Arguments, Mathematics.
- Basics of File Handling
- Introduction to Libraries



COMING UP NEXT:



LETS BUILD A CALCULATOR !

Join Here!

Thank you for joining me!

Looking forward to seeing you guys next week!