Learning to Speak Python 2

June 12, 2021

Starting on a Soapbox

1011 one bit 1 byte = 8 bits

$1011_2 = ?_{10}$

$\underset{3}{1}\ \underset{2}{0}\ \underset{1}{1}\ \underset{0}{1} = 2^0 + 2^1 + 2^3$

$= 1 + 2 + 8$

$1011 = 11_{10}$

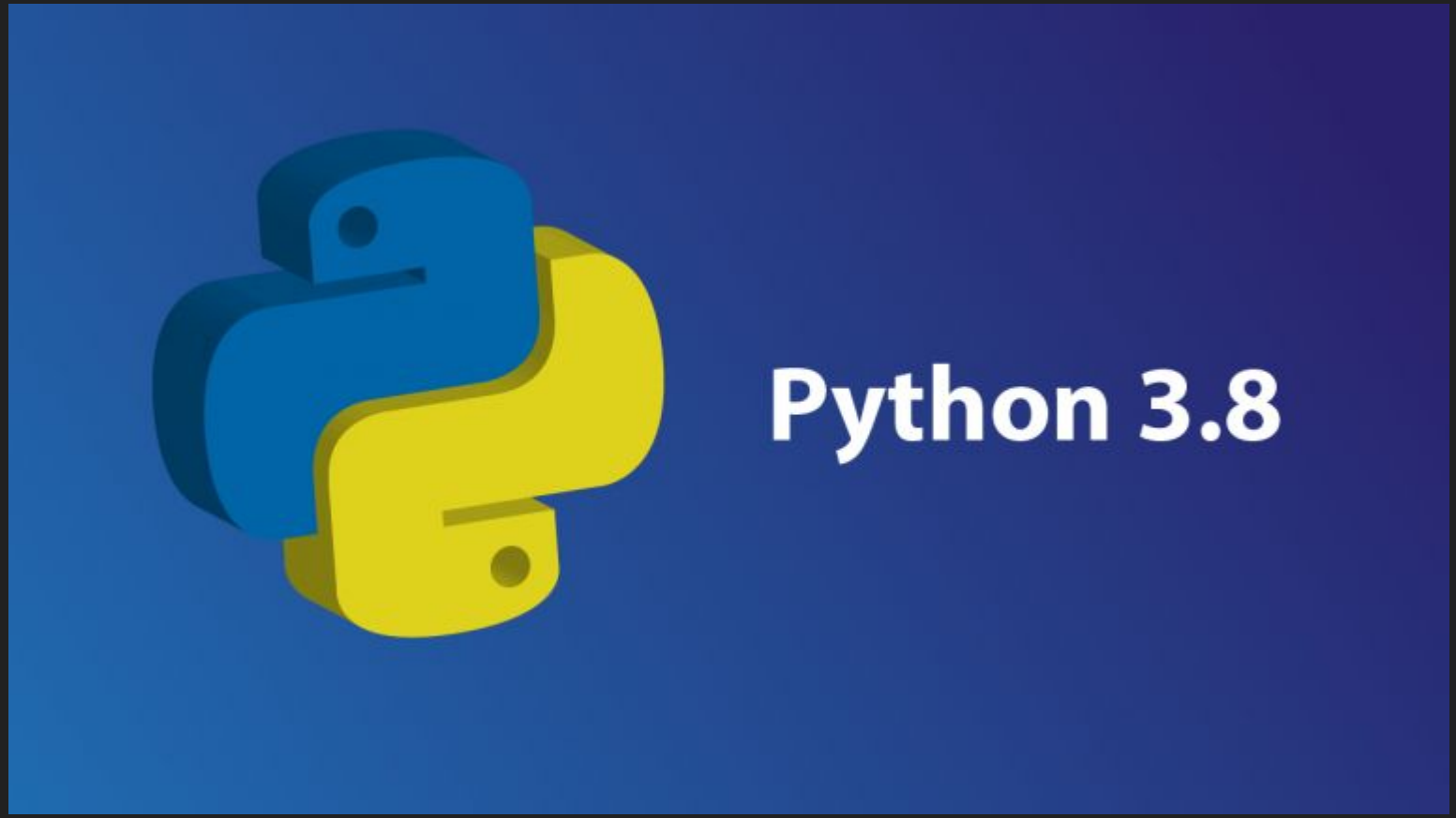

MFW
MACHINE CODE


MFW
ALLOCATING BYTES TO MEMORY


MFW
I SUFFER TO DO BASIC
COMPUTATION ON MY COMPUTER


OH MY GOD


OFW
HIGH-LEVEL PROGRAMMING LANGUAGE


Python 3.8

Roadmap for the day ^^

Review

data-types
indexing
for loops

Practice

Print out index, val of
[“a” , “b” ,” c”]

Introduce

math, % // * ** - +

Practice

How can we check if
a number is odd?

Practice

Explanations

Totally optional

Introduce

boolean logic
if statements
while loops (&
break!)

Practice

Can we loop through
a range of numbers
less than 10, and
print out odd?

Introduce

functions
return , and yield

Practice

Can you return 3
odd numbers above
and below an input?

Practice

3s and 5s
Fibonacci

Introduce

looping through
dictionaries

Practice

Can you give me the
dictionary value of
an input key?

Practice

Cash register that
gives you change

Writing bubble sort

Review

data-types
indexing
for loops

Practice

Print out index, val of
[“a” , “b” ,” c”]

Introduce

math, % // * ** - +

Practice

How can we check if
a number is odd?

Introduce

boolean logic
if statements
while loops (&
break!)

Practice

Can we loop through
a range of numbers
less than 10, and
print out odd?

Introduce

functions
return , and yield

Practice

Can you return 3
odd numbers above
and below an input?

Practice

3s and 5s
Fibonacci

Introduce

looping through
dictionaries

Practice

Can you give me the
dictionary value of
an input key?

Practice

Cash register that
gives you change

Writing bubble sort

```
["a","b","c"]
[1, 2, 2]
"abc123"
{1, 2, 3}
{ 'nihao': 'hello'}
```

```
set([1, 1])
```

```
[“a”,”b”,”c”]
[1, 2, 2]
“abc123”
{1, 2, 3}
{ ‘nihao’: ‘hello’}
```

```
set([1, 1])  #Output: {1}
```

```
["a","b","c"]
[1, 2, 2]
"abc123"
{1, 2, 3}
{ 'nihao': 'hello'}
```

```
set((1, 1))
```

```
[“a”,”b”,”c”]
[1, 2, 2]
“abc123”
{1, 2, 3}
{ ‘nihao’: ‘hello’}
```

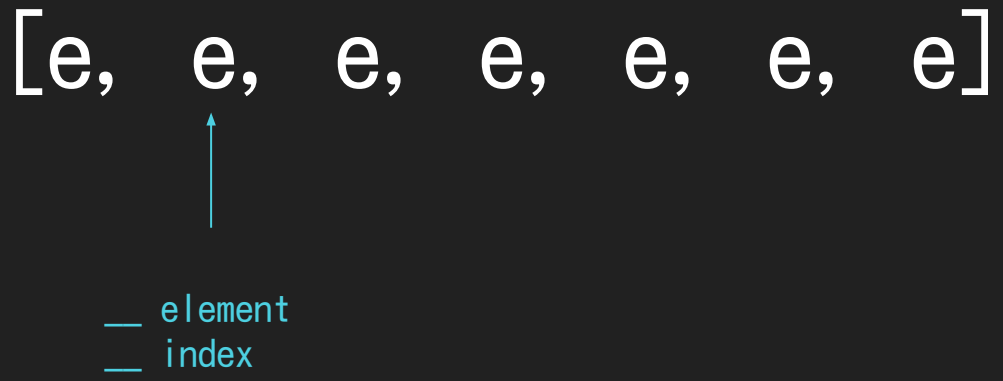
```
set((1, 1))  #Output: {1}
```

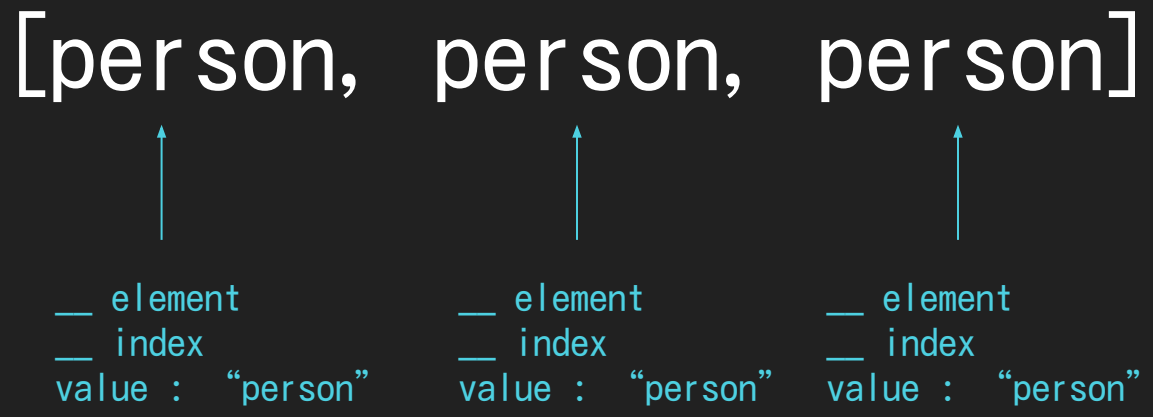
```
["a","b","c"]
[1, 2, 2]
"abc123"
{1, 2, 3}
{ 'nihao': 'hello'}
```

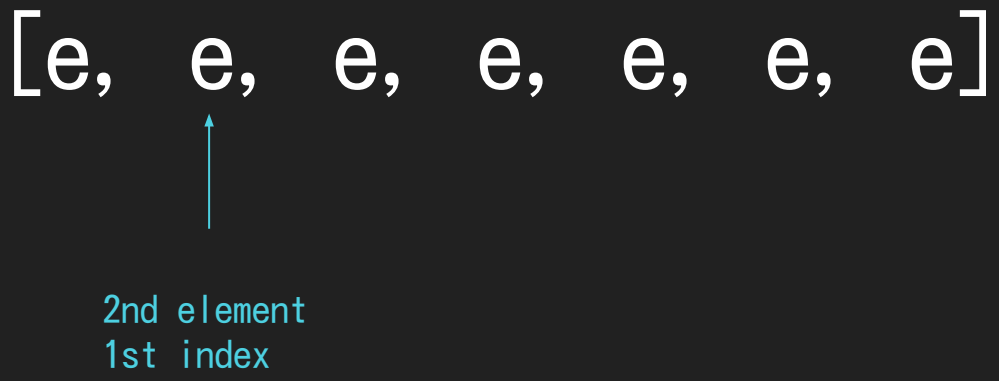
```
list((1,1))
```

```
["a","b","c"]
[1, 2, 2]
"abc123"
{1, 2, 3}
{ 'nihao': 'hello'}
```

```
list((1,1))  #Output: [1, 1]
```

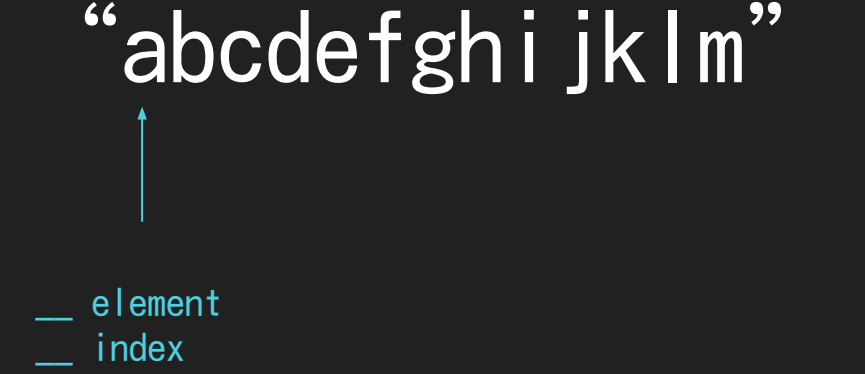

[e, e, e, e, e, e, e]
element
index


[person, person, person]
__ element
__ index
value : “person”
__ element
__ index
value : “person”
__ element
__ index
value : “person”


[e, e, e, e, e, e, e]
2nd element
1st index


__ element
__ index
[e, e, e, e, e, e, e]


7th element
6th index
[e, e, e, e, e, e, e]


“abcdefghijklm”
__ element
__ index

“abcdefghijklm”

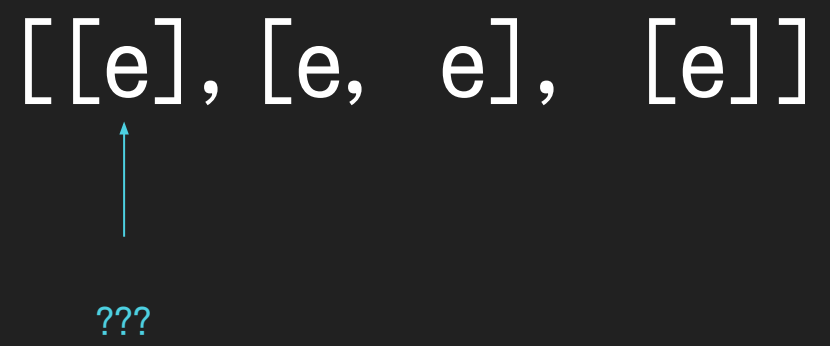
1st element
0th index
value: ‘a’

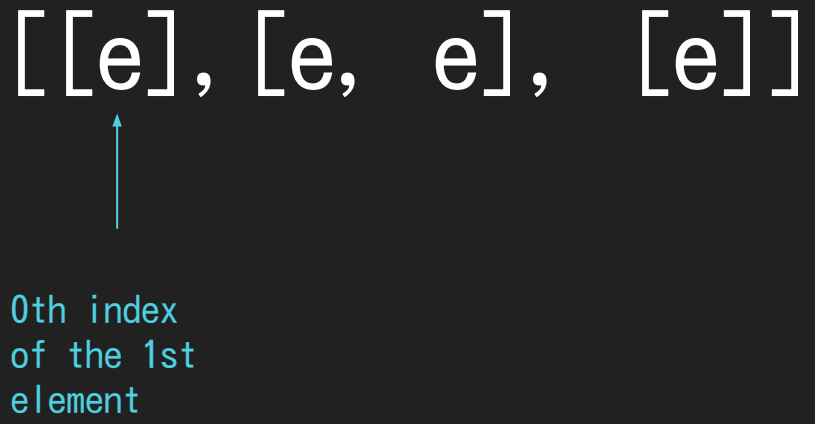
“abcdefghijklm”

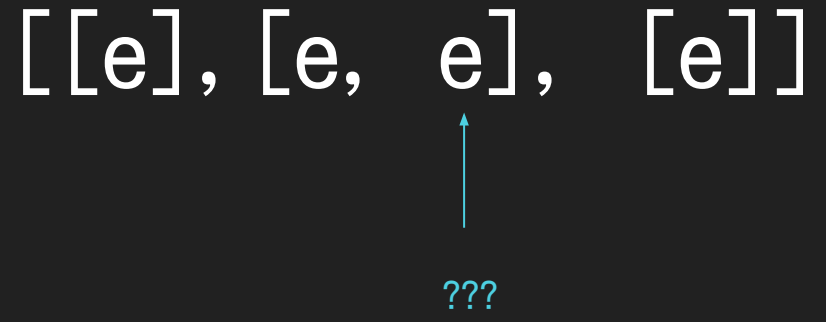
__ element
__ index

“abcdefghijklm”

4th element
3rd index


[[e], [e, e], [e]]
???


[[e], [e, e], [e]]
0th index
of the 1st
element


[[e], [e, e], [e]]
???

[[e], [e, e], [e]]

1st index
of the 1st
element

Whiteboard this for woop.

```
bean_count = 0
the_jar = [ ‘bean’ , ‘bean’ , ‘bean’ , ‘bean’ ]

for bean in the_jar:
    bean_count = bean_count + 1

print(f ‘There are { bean_count = } in the jar!’ )
```

Review

data-types
indexing
for loops

Practice

Print out index, val of
[“a” , “b” ,” c”]

Introduce

math, % // * ** - +

Practice

How can we check if
a number is odd?

Practice

Explanations

Totally optional

Introduce

boolean logic
if statements
while loops (&
break!)

Practice

Can we loop through
a range of numbers
less than 10, and
print out odd?

Introduce

functions
return , and yield

Practice

Can you return 3
odd numbers above
and below an input?

Practice

3s and 5s
Fibonacci

Introduce

looping through
dictionaries

Practice

Can you give me the
dictionary value of
an input key?

Practice

Cash register that
gives you change

Writing bubble sort

Let’s assume I have a list ‘a’, ‘b’, ‘c’ and want to, for each element in the list, print out the given element’s index, value.

Example output:
Val: ‘a’ Index: 0
Val: ‘b’ Index: 1
Val: ‘c’ Index: 2

```
l = [ ‘a’ , ‘b’ , ‘c’ ]

for i in l:
    print(i)

#Output: a, b, c
```

```
l   = [ ‘a’ , ‘b’ , ‘c’ ]
ind = 0

for i in l:
    print(i, ind)
    ind = ind + 1

#Output:
a 0
b 1
c 2
```

```
l   = [ ‘a’ , ‘b’ , ‘c’ ]

for index, value in enumerate(l):
    print(index, value)

#Output:
0 a
1 b
2 c
```

Review

data-types
indexing
for loops

Practice

Print out index, val of
[“a” , “b” ,” c”]

Introduce

math, % // * ** - +

Practice

How can we check if
a number is odd?

Introduce

boolean logic
if statements
while loops (&
break!)

Practice

Can we loop through
a range of numbers
less than 10, and
print out odd?

Introduce

functions
return , and yield

Practice

Can you return 3
odd numbers above
and below an input?

Practice

3s and 5s
Fibonacci

Introduce

looping through
dictionaries

Practice

Can you give me the
dictionary value of
an input key?

Practice

Cash register that
gives you change

Writing bubble sort

Practice

Explanations

Totally optional

[1]:
```
l = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
for i in l:
    print(i%2)
```

```
1
0
1
0
1
0
1
0
1
0
```

What the heck you think is happening here, huh?

[10]:
```
l = [10, 100, 200]
for i in l:
    print(i//2)
```

```
5
50
100
```

What's goin on...

Review

data-types
indexing
for loops

Practice

Print out index, val of
[“a” , “b” ,” c”]

Introduce

math, % // * ** - +

Practice

How can we check if
a number is odd?

Practice

Explanations

Totally optional

Introduce

boolean logic
if statements
while loops (&
break!)

Practice

Can we loop through
a range of numbers
less than 10, and
print out odd?

Introduce

functions
return , and yield

Practice

Can you return 3
odd numbers above
and below an input?

Practice

3s and 5s
Fibonacci

Introduce

looping through
dictionaries

Practice

Can you give me the
dictionary value of
an input key?

Practice

Cash register that
gives you change

Writing bubble sort

Odd Numbers Chart

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

Review

data-types
indexing
for loops

Practice

Print out index, val of
[“a” , “b” ,” c”]

Introduce

math, % // * ** - +

Practice

How can we check if
a number is odd?

Practice

Explanations

Totally optional

Introduce

boolean logic
if statements
while loops (&
break!)

Practice

Can we loop through
a range of numbers
less than 10, and
print out odd?

Introduce

functions
return , and yield

Practice

Can you return 3
odd numbers above
and below an input?

Practice

3s and 5s
Fibonacci

Introduce

looping through
dictionaries

Practice

Can you give me the
dictionary value of
an input key?

Practice

Cash register that
gives you change

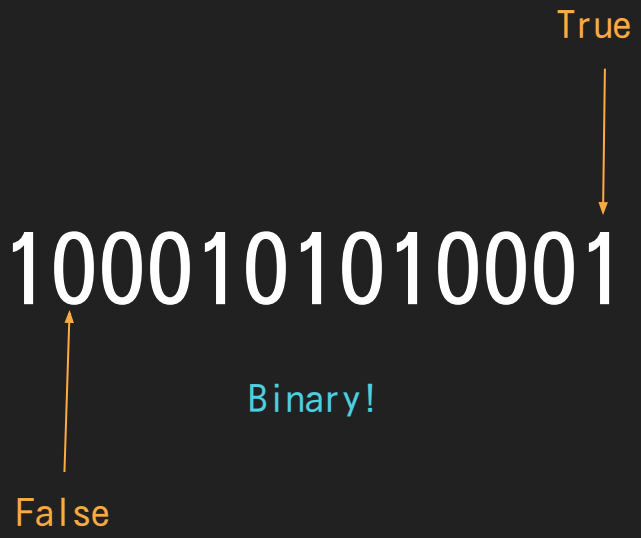
Writing bubble sort

1000101010001

What is this?

1000101010001

Binary!

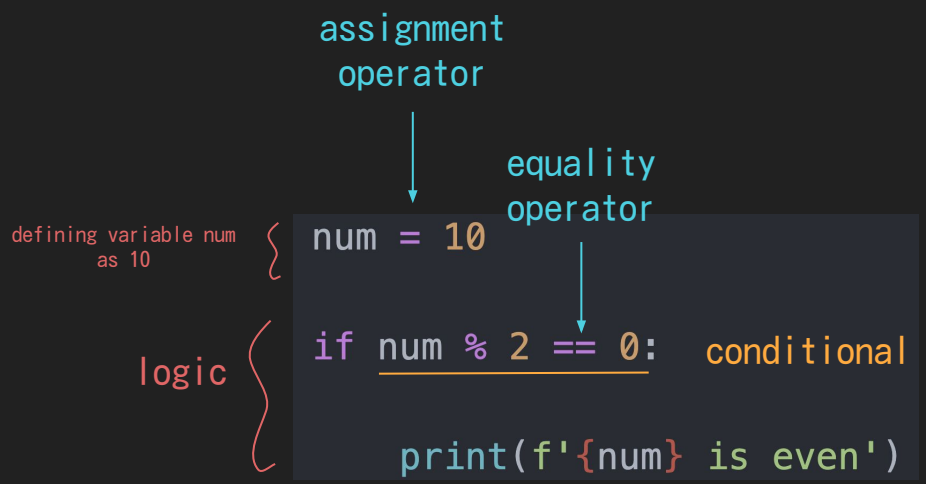

True
1000101010001
Binary!
False

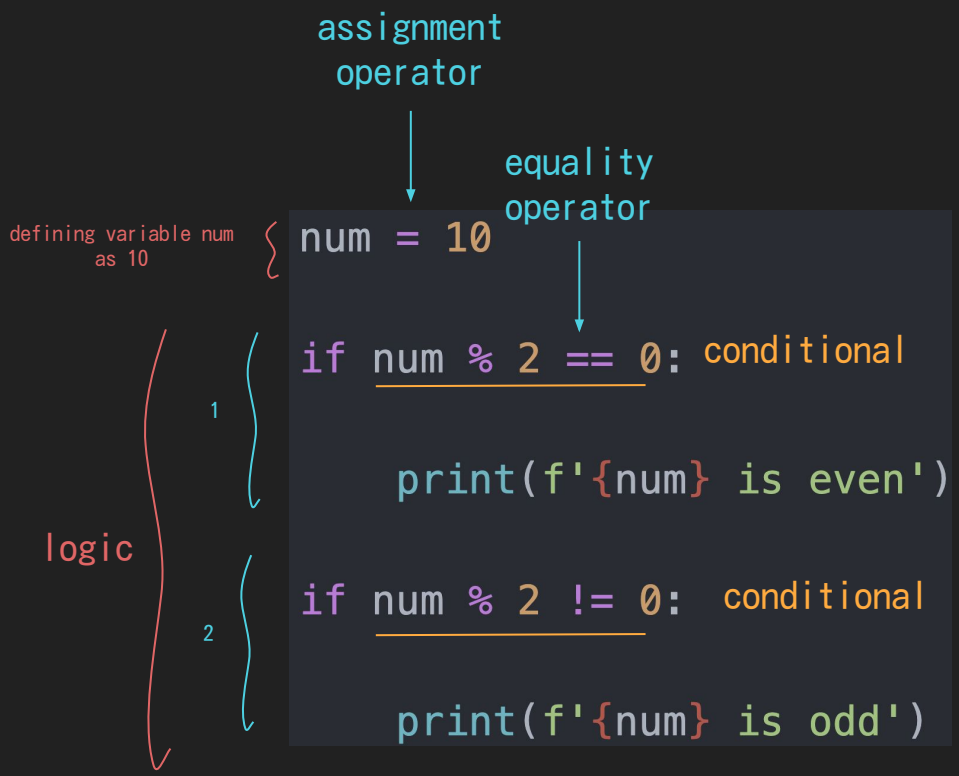
A computer thinks as “yes” or “no”. Life, however, is not like this -- but we must adapt this mentality anyway.

If statements

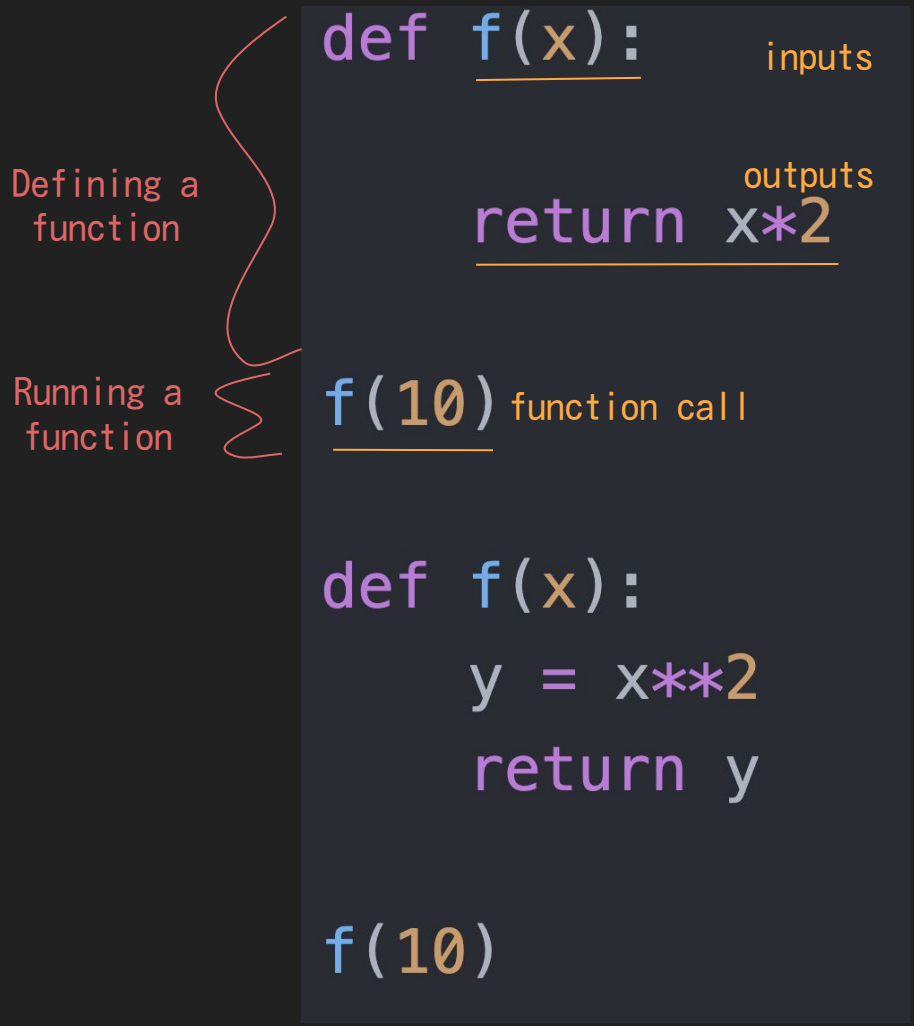
How would we detect if a number is odd?

```
if …
   do something
```


assignment operator
equality operator
defining variable num as 10
num = 10
logic
if num % 2 == 0:
conditional
print(f'{num} is even')


assignment operator
equality operator
defining variable num as 10
num = 10
if num % 2 == 0: conditional
print(f'{num} is even')
if num % 2 != 0: conditional
print(f'{num} is odd')
logic
1
2

functions


def f(x):
inputs
outputs
return x*2
Defining a function
Running a function
f(10)
function call
def f(x):
y = x**2
return y
f(10)