Primitives
Data types

TECHNION
Azrieli Continuing Education and External Studies Division

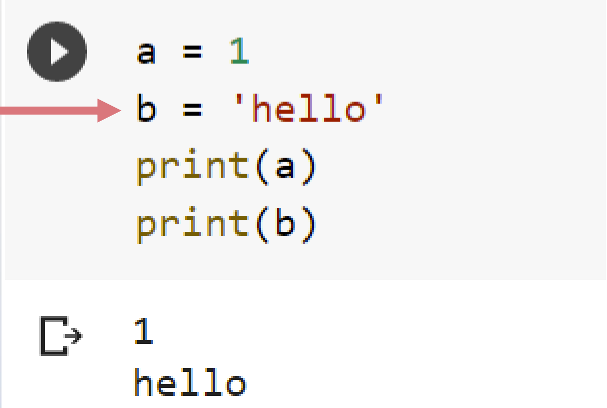
Agenda

1. Variables.
2. Numbers – int and float.
3. Boolean.
4. Text – string and char.

Variables

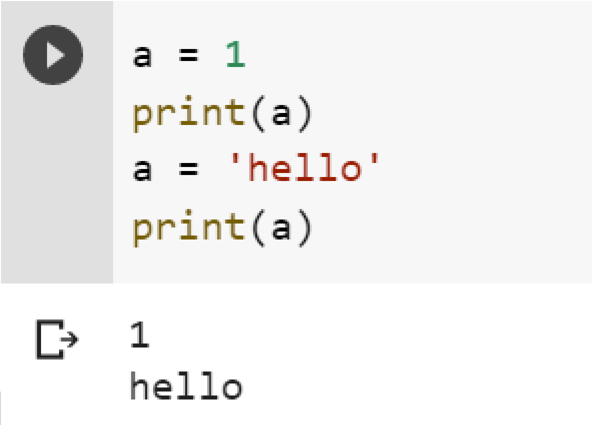
- Variable - a container for storing data values (number, text, list etc.).
- Variable assignment: `variable_name = variable_value`

No type needed



```
a = 1
b = 'hello'
print(a)
print(b)
```

```
1
hello
```



```
a = 1
print(a)
a = 'hello'
print(a)
```

```
1
hello
```

Naming convention

- Variables:

  All small letters.
  Separate words with _.
  Don’t start with a number.
  Don’t use keywords

  * Don’t refer to non defined variables

Variables

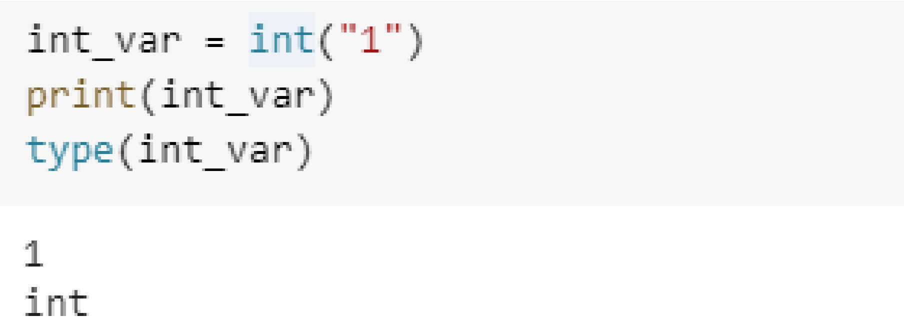
Primitives (singular)	Non- Primitives(Collections)
Integer	List[]
Float	Dictionary{}
String	Tuple()
Boolean	Set{}
By Python	By Programmer

type(variable_name)→ returns the data type of the variable

Convert between types

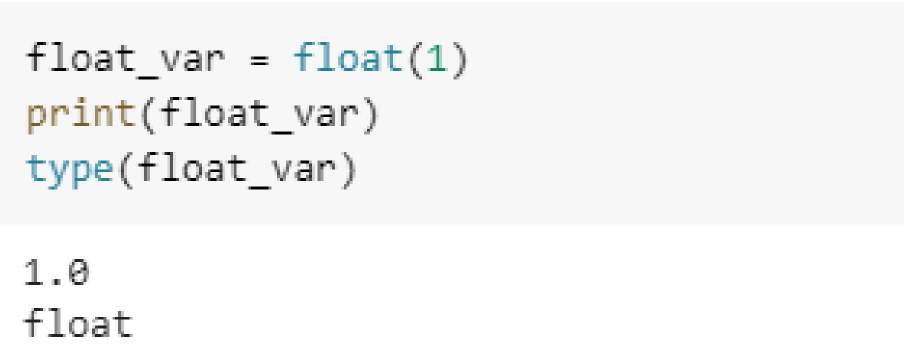
- type() – check the variable type.
- Convert between types using the functions

  int()
  str()
  float().



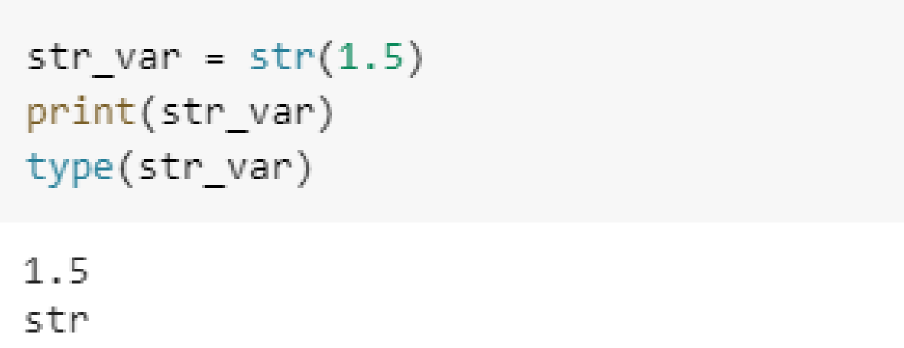
```
int_var = int("1")
print(int_var)
type(int_var)
```

```
1
int
```



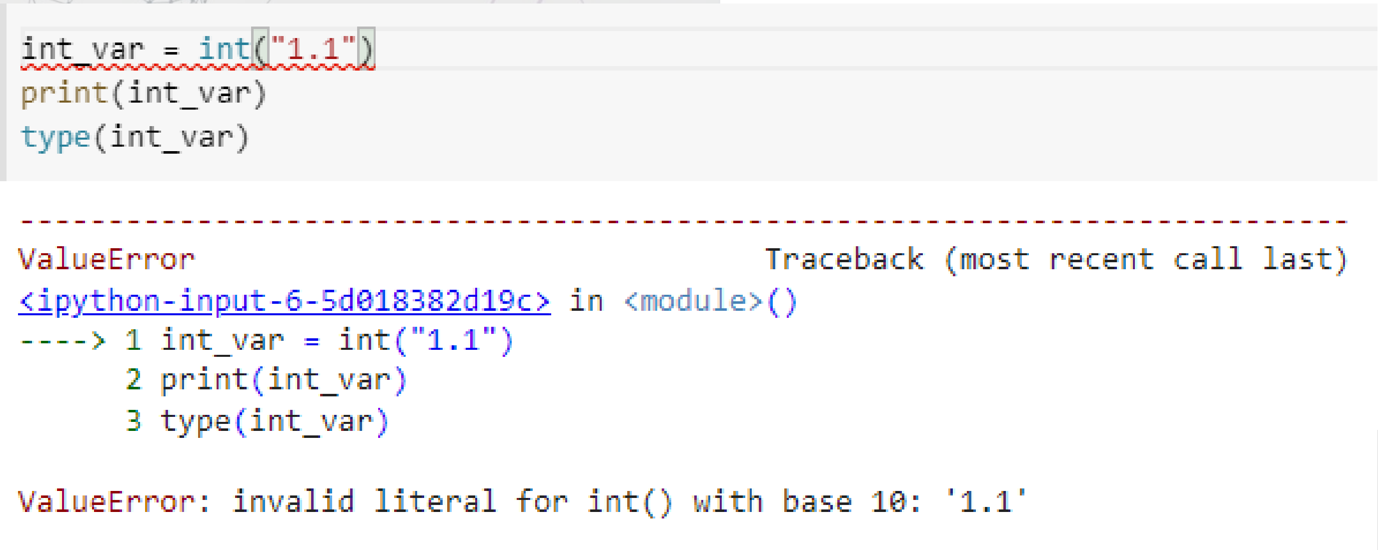
```
float_var = float(1)
print(float_var)
type(float_var)
```

```
1.0
float
```



```
str_var = str(1.5)
print(str_var)
type(str_var)
```

```
1.5
str
```



```
int_var = int("1.1")
print(int_var)
type(int_var)
```

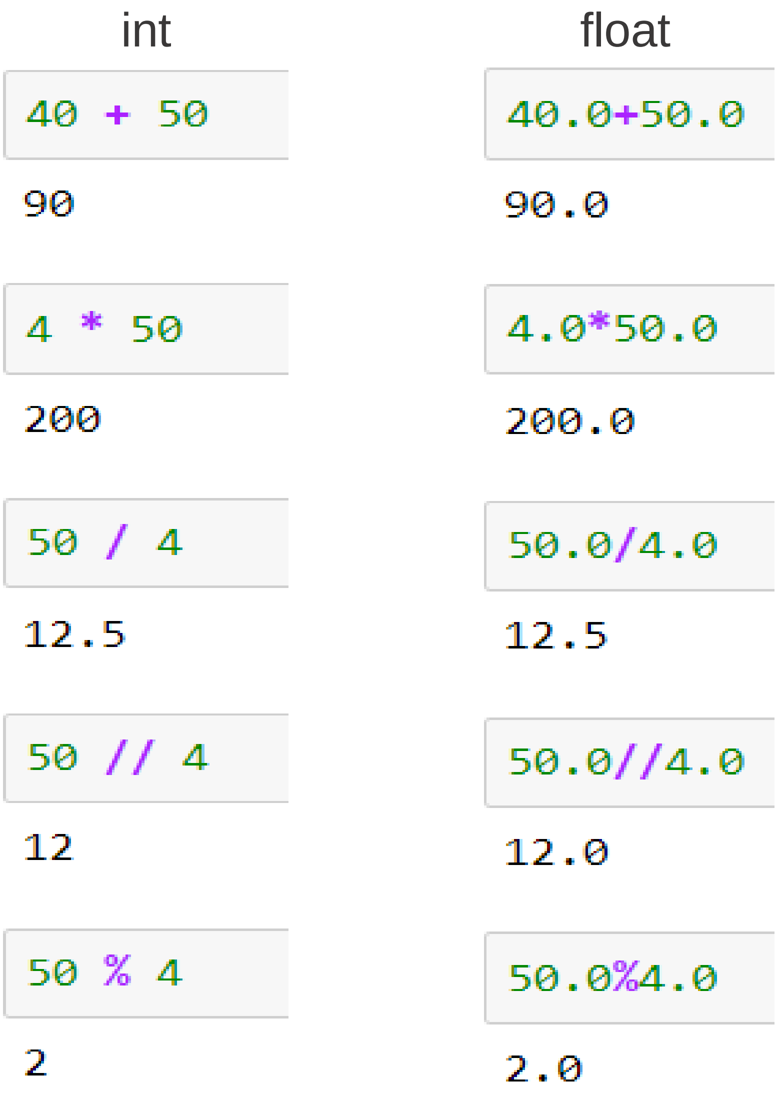
```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-6-5d018382d19c> in <module>()
----> 1 int_var = int("1.1")
      2 print(int_var)
      3 type(int_var)

ValueError: invalid literal for int() with base 10: '1.1'
```

Numbers

Numbers

Operator	Name
+	Addition
-	Subtraction
*	Multiplication
/	Division
%	Modulus [1]
**	Exponentiation
//	Floor division

[1] the remainder of a division.



int

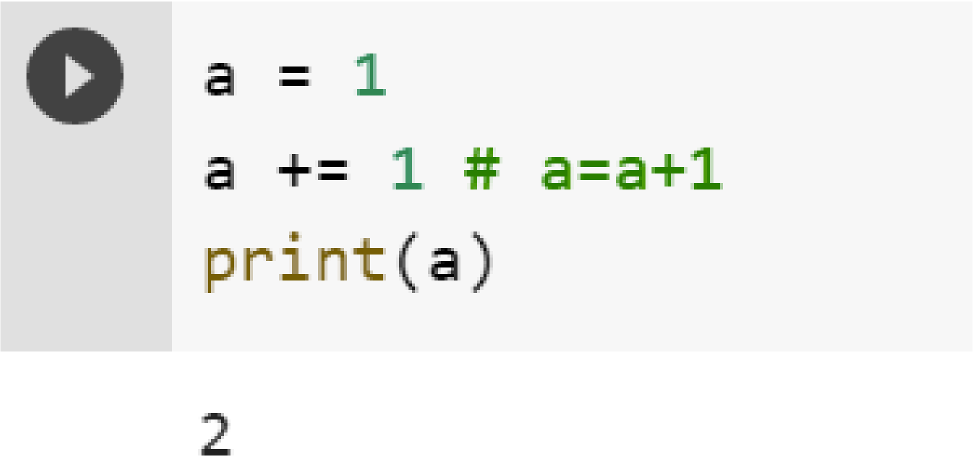
```
40 + 50
90

4 * 50
200

50 / 4
12.5

50 // 4
12

50 % 4
2
```

float

```
40.0+50.0
90.0

4.0*50.0
200.0

50.0/4.0
12.5

50.0//4.0
12.0

50.0%4.0
2.0
```

Numbers

- Assignment with an operation:
  - **+=**
  - **-=**
  - ***=**
  - **/=**



```
a = 1
a += 1 # a=a+1
print(a)
```

```
2
```

Exercise

1. **Declare x and y variables to 8 and -5.**
2. **Print the sum of x and y.**
3. **Print the difference between x and y.**
4. **Print the multiplication between x and y.**
5. **Add y to x and print x.**
6. **Set y to be equal to y squared and print y.**
7. **Calculate:**
    1. **X // 3**
    2. **X % 3**
    3. **X * 0.5**
    4. **X ** 2**

Bonus:

- **What type is the result of 10/3?**
- **What type is the result of 10//3?**

1815-1864

Boolean

Boolean

- Can hold two values: True or False.

Operator	Name	Example	Result

==	Equal	x == 3	x = 3 → True
			x = 7 → False

!=	Not equal	x != 3	x = 3 → False
			x = 7 → True

>	Greater than [1]	x > 3	x = 3 → False
			x = 7 → True

>=	Greater than [1] or equal to	x >= 3	x = 3 → True
			x = 7 → True

[1] use < for smaller than

Boolean

- Logical operators:

Operator	Name	Example	Result

and	Returns True if both statements are true	x < 5 and x < 10	x = 3 → True
			x = 7 → False

or	Returns True if one of the statements is true	x < 5 or x < 10	x = 3 → True
			x = 7 → True

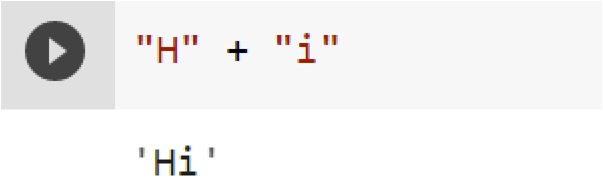
not	Reverse the result, returns False if the result is true	not(x < 5 and x < 10)	x = 3 → False
			x = 7 → True

Exercise

Text

Char

- A single character in text, e.g. “A”.
- Not very useful but good to know.
- Adding chars together creates a

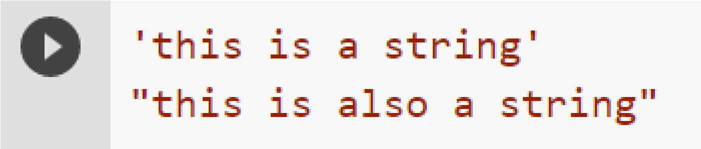


```
"H" + "i"
```

```
'Hi'
```

String

- Represent a list of characters.
- Can be defined with ‘ ‘ or “ “.

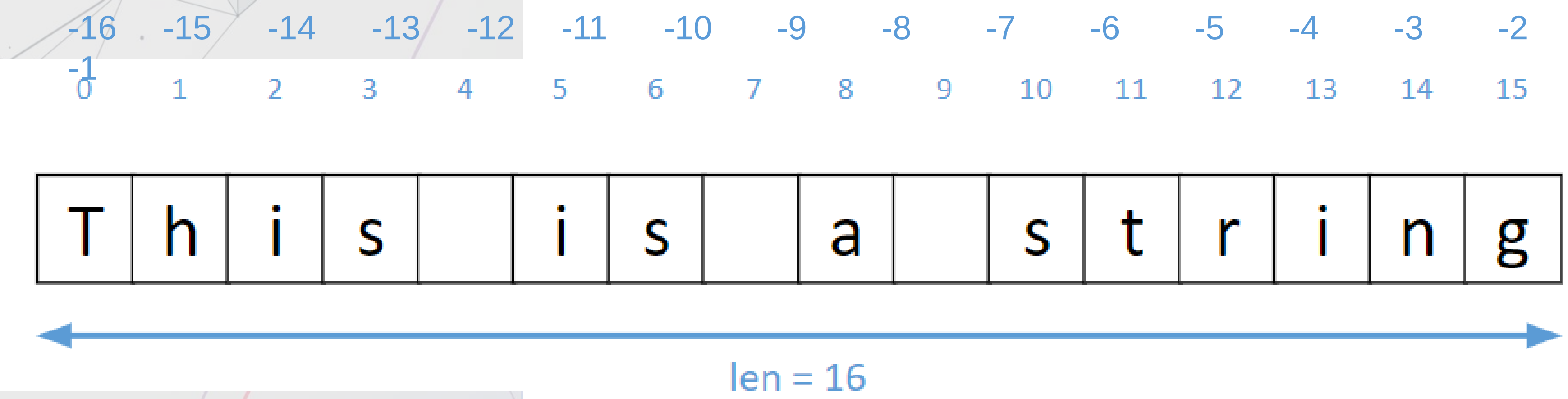


```
'this is a string'
"this is also a string"
```

String slicing

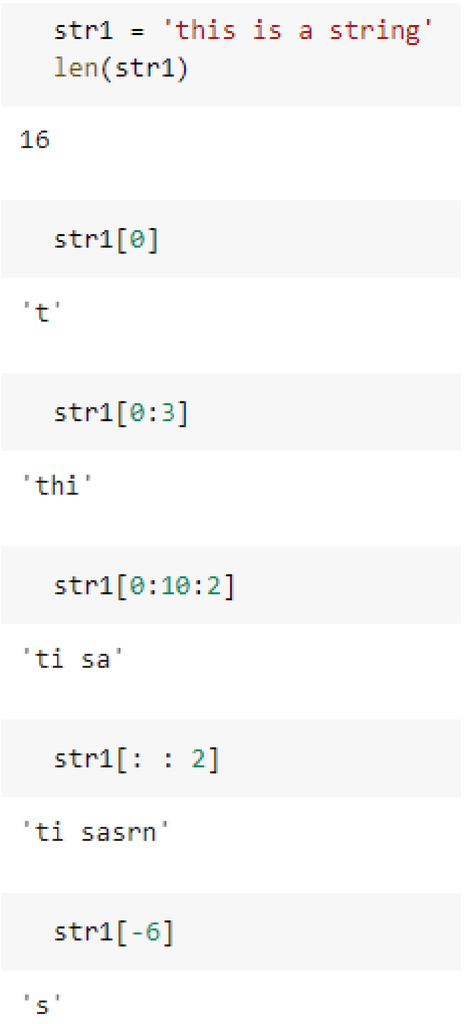
- Index start at zero.
- Slicing

```
str[start : stop : steps]
```



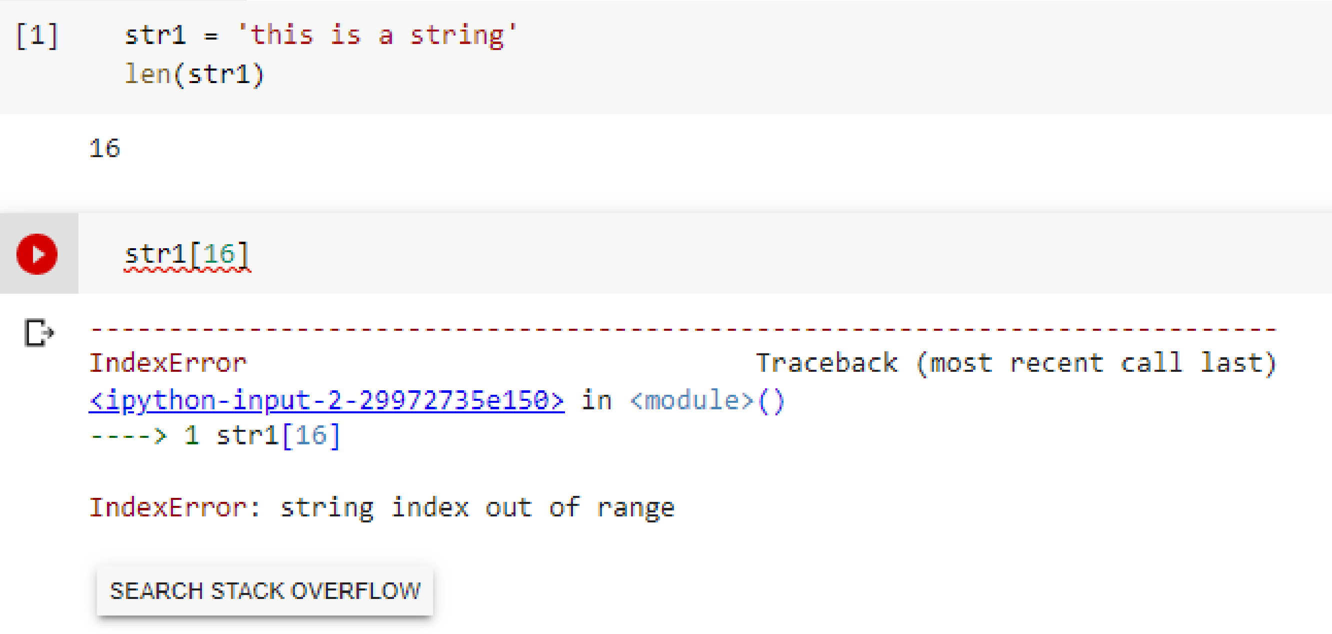
-16 -1	-15	-14	-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
T	h	i	s		i	s		a		s	t	r	i	n	g

len = 16



```
str1 = 'this is a string'
len(str1)

16

str1[0]

't'

str1[0:3]

'thi'

str1[0:10:2]

'ti sa'

str1[: : 2]

'ti sasrn'

str1[-6]

's'
```

String slicing



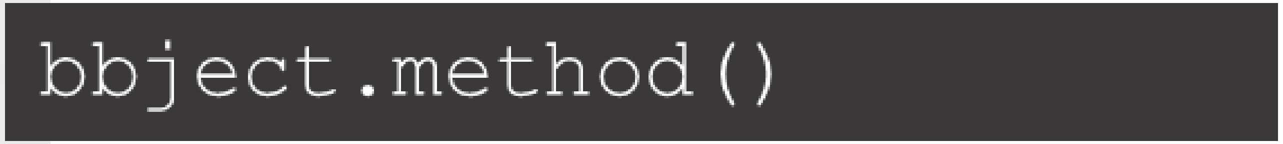
```
[1]  str1 = 'this is a string'
     len(str1)

16
```

```
str1[16]
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
<ipython-input-2-29972735e150> in <module>()
----> 1 str1[16]

IndexError: string index out of range
```

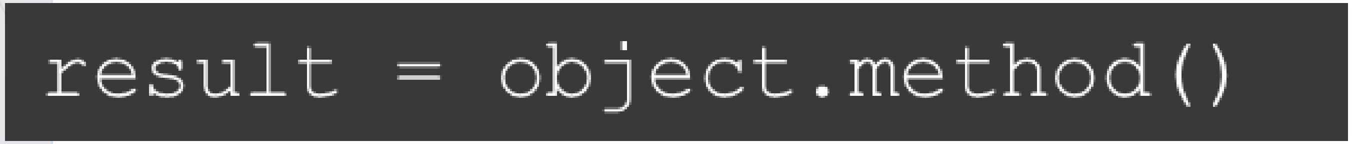
String methods

- String is an object (more about objects later in the course).
- Objects can have methods.
- Call a method:



```
bbject.method()
```

- All string methods returns new values, they do not change the original string.



```
result = object.method()
```

String methods

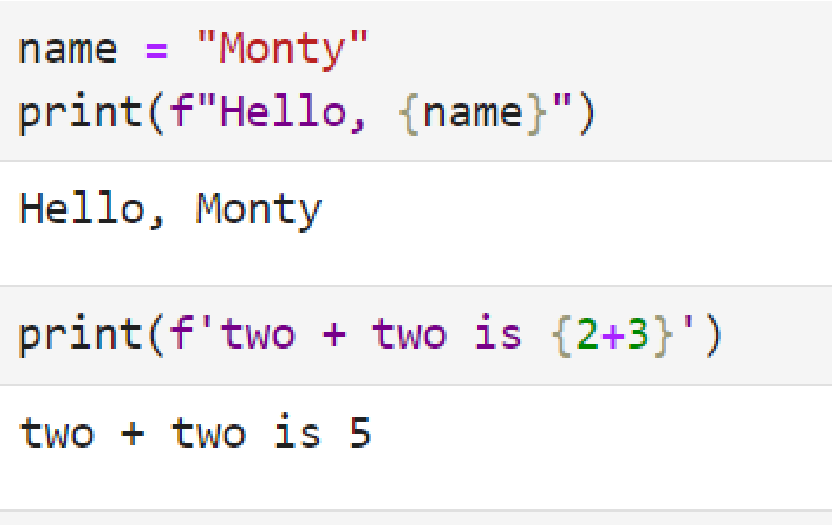
- **Find()** – return the index of a substring.
- **Replace()** – replace a substring with another string.
- **Count()** – return the number of times a substring occurs.
- **Upper()** – return the same string but with all upper-case letters.
- And more…

Special strings

- **f-string syntax:**

  Easley format strings.

  Evaluated at runtime, so you can add any function or calculation in { }.



```
name = "Monty"
print(f"Hello, {name}")
```

```
Hello, Monty
```

```
print(f'two + two is {2+3}')
```

```
two + two is 5
```

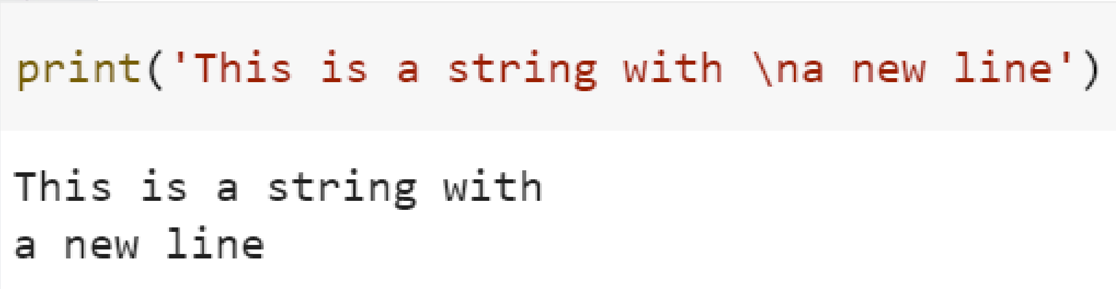
- **Special characters in**

  \n – new line

  \t – tab

  \\ - backslash

  \’ – single quote



```
print('This is a string with \na new line')
```

```
This is a string with
a new line
```

Exercise

Exercise

1. Declare variable named “dollar” as a number your choosing (for example 32).
2. Print the conversion of dollar to shekels (1 dollar = 3.72 shekel) in the following format:
3. 32$ is equal to 29.59 NIS.
4. Declare a string as “no one can solve this” and replace the words “solve this” with your name.
5. Declare str1 as “100+200”.
6. Cut str1 into two strings (s_num1, s_num2), each consist only with one of the nembers.
7. Convert s_num1, s_num2 into numbers num1, num2.
8. Print:
    1. str1
    2. s_num1 + s_num2
    3. num1 + num2

Summary

Variable

a container for storing data values.

Numbers

Basic arithmetic operations: +, -, * etc.

Boolean

True / False, logical operations.

Text

Strings slicing, methods and special strings.

Convert types

int(), float(), str(), type().

Questions?