



# 04. STRINGS AND NUMBERS

M2U5P4

The background is a dark blue gradient. In the corners, there are white line-art illustrations of circuit boards or neural networks, consisting of lines and small circles.

# WHAT ARE STRINGS?

# PYTHON STRINGS

- Strings is a textual type of data
- It can contain 1 character, 1 word or more words
- String in Python are surrounded by either single quotation marks, or double quotation marks.

```
print("Hello")  
print('Hello')
```

# ASSIGN STRING TO A VARIABLE

- Assigning a string to a variable is done with the variable name followed by an equal sign and the string:

```
a = "Hello"  
print(a)
```

# MULTILINE STRINGS

- We can assign a multiline string to a variable by using three quotes or three single quotes:

```
a = """Lorem ipsum dolor sit amet,  
consectetur adipiscing elit,  
sed do eiusmod tempor incididunt  
ut labore et dolore magna aliqua."""  
print(a)
```

```
a = '''Lorem ipsum dolor sit amet,  
consectetur adipiscing elit,  
sed do eiusmod tempor incididunt  
ut labore et dolore magna aliqua.'''  
print(a)
```

# STRING LENGTH

- We can measure the length of a string by using the `len()` function

```
a = "Hello, World!"  
print(len(a))
```

# COMBINING STRINGS

- We can combine two or more strings into one string by using **+** operator

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

# COMBINING STRINGS

- We can combine two or more strings into one string by using **+** operator

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

The image features a dark blue gradient background with white, stylized circuit board traces in the corners. These traces consist of straight lines and small circles, resembling electronic components or data paths. The traces are located in the top-left, top-right, bottom-left, and bottom-right corners, framing the central text.

ANY QUESTIONS ABOUT STRINGS?

The background is a solid teal color with a subtle gradient. In the four corners, there are decorative white line-art patterns resembling circuit traces or neural network connections. These patterns consist of straight lines of varying lengths and angles, ending in small white circles. The patterns are most prominent in the top-left and bottom-left corners, and less so in the top-right and bottom-right corners.

WHAT ARE NUMBERS?

# NUMBERS

- A number is mathematical object used to count, measure and label things
- Calculations with numbers are done with arithmetical operations
- Most familiar operations are addition, subtraction, multiplication and division
- What is the most highest number?
- What is the most lowest number?

# PYTHON NUMBERS

- There are three numeric types in Python
  - `int`
  - `float`
  - `complex`

```
x = 1      # int
y = 2.8    # float
z = 1j     # complex
```

# INT

- Int, or integer, is a whole number, positive or negative, without decimals, of unlimited length

```
x = 1  
y = 35656222554887711  
z = -3255522
```

```
print(type(x))  
print(type(y))  
print(type(z))
```

# FLOAT

- Float, or "floating point number" is a number, positive or negative, containing one or more decimals

```
x = 1.10  
y = 1.0  
z = -35.59
```

```
print(type(x))  
print(type(y))  
print(type(z))
```

# TYPE CONVERSION

- We can convert from one type to another with the `int()` and `float()` methods:

```
1.0
2
<class 'float'>
<class 'int'>
```

```
x = 1    # int
y = 2.8  # float
```

```
#convert from int to float:
a = float(x)
```

```
#convert from float to int:
b = int(y)
```

```
print(a)
print(b)
```

```
print(type(a))
print(type(b))
```

# RANDOM NUMBER

- Some programming languages have a built in `random()` function to make a random number
- Python doesn't have it....
- Python has a built-in module called `random` that can be used to make random numbers:

```
import random  
  
print(random.randrange(1, 34))
```

The background is a gradient of blue, transitioning from a lighter shade at the top to a darker shade at the bottom. In the four corners, there are decorative white line-art elements resembling circuit traces or neural network connections, with small circles at the end of the lines.

ANY QUESTIONS ABOUT NUMBERS?

The background is a gradient of blue, transitioning from a lighter shade at the top to a darker shade at the bottom. In the four corners, there are decorative white line-art elements resembling circuit traces or neural network connections, with small circles at the end of the lines.

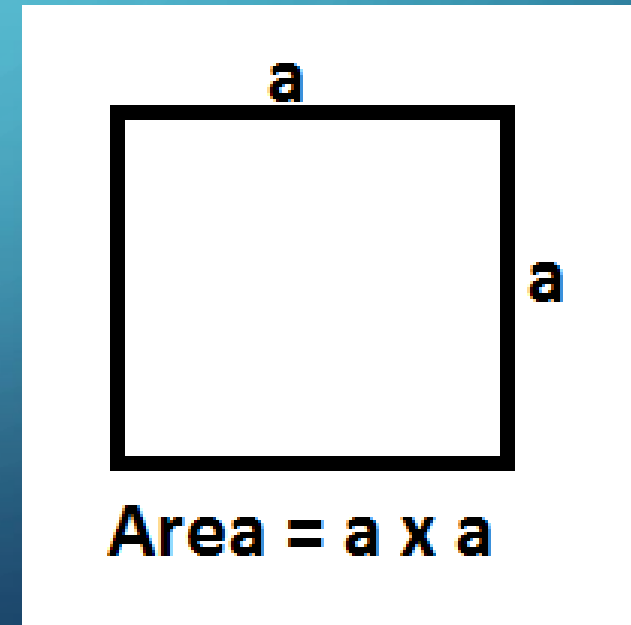
# CLASSROOM WORK TIME

# TASK 1: AREA OF SQUARE

- Make a program that will calculate the area of a square
- Ask for input from the user
- Calculate it and give the result

# TASK 1: AREA OF SQUARE

- Make a program that will calculate the area of a square
- Ask for input from the user
- Calculate it and give the result



## TASK 1: AREA OF SQUARE

```
1 print("This program will calculate the area of a square")
2 a = int(input("Enter the length of the side: "))
3 area = a * a
4 print("Area of the square is " + area)
```

# TASK 1: AREA OF SQUARE

## DOES IT WORK?

```
1 print("This program will calculate the area of a square")
2 a = int(input("Enter the length of the side: "))
3 area = a * a
4 print("Area of the square is " + area)
```

## TASK 1: AREA OF SQUARE

NO IT DOESN'T

```
1 print("This program will calculate the area of a square")
2 a = int(input("Enter the length of the side: "))
3 area = a * a
4 print("Area of the square is " + area)
```

Traceback (most recent call last):

File "<pyshell>", line 4, in <module>

TypeError: can only concatenate str (not "int") to str

## TASK 1: AREA OF SQUARE

```
1 print("This program will calculate the area of a square")
2 a = int(input("Enter the length of the side: "))
3 area = a * a
4 print("Area of the square is ", area)
```

# STRING + NUMBER

- As we learned before that we cannot combine strings and numbers like this:

```
age = 36  
txt = "My name is John, I am " + age  
print(txt)
```

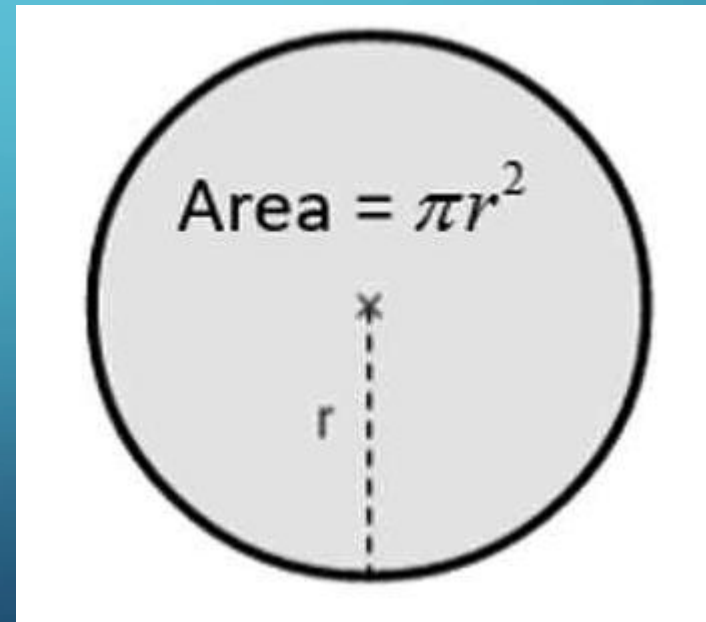
# STRING + NUMBER

- If we want to combine them together, we should use a **comma (,)** sign:

```
1 age = 36
2 print("My name is John, and I am", age , "years old")
```

## TASK 2: AREA OF CIRCLE

- Make a program that will calculate the area of a circle
- Ask for input from user
- Calculate it and give result

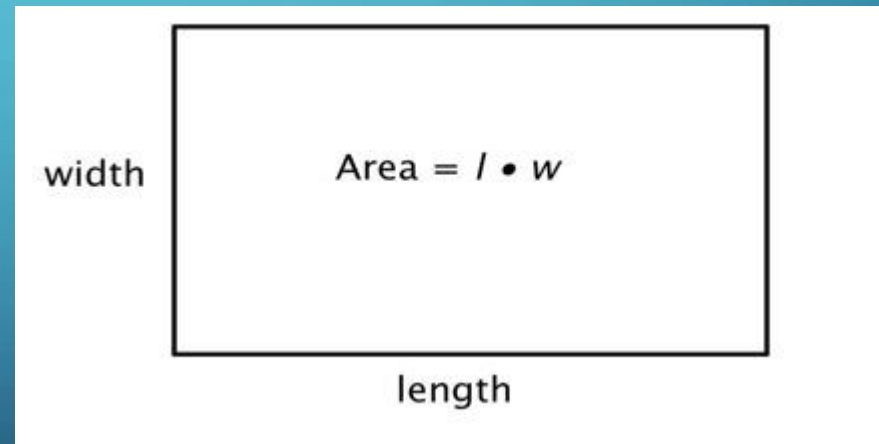


## TASK 2: AREA OF CIRCLE

```
1 print("This program will calculate the area of a circle")
2 r = int(input("Enter the radius of the circle: "))
3 area = 3.14 * r * r
4 print("Area of the circle is ", area)
```

## TASK 3: RECTANGLE AREA

- Make a program that will calculate the area of a rectangle
- Ask for input from user
- Calculate it and give result



## TASK 3: RECTANGLE AREA

```
1 print("This program will calculate the area of a rectangle")
2 l = int(input("Enter the length of the rectangle: "))
3 w = int(input("Enter the width of the rectangle: "))
4 area = w * l
5 print("Area of the rectangle is ", area)
```



ANY QUESTIONS?



# WORKSHEET



THE END