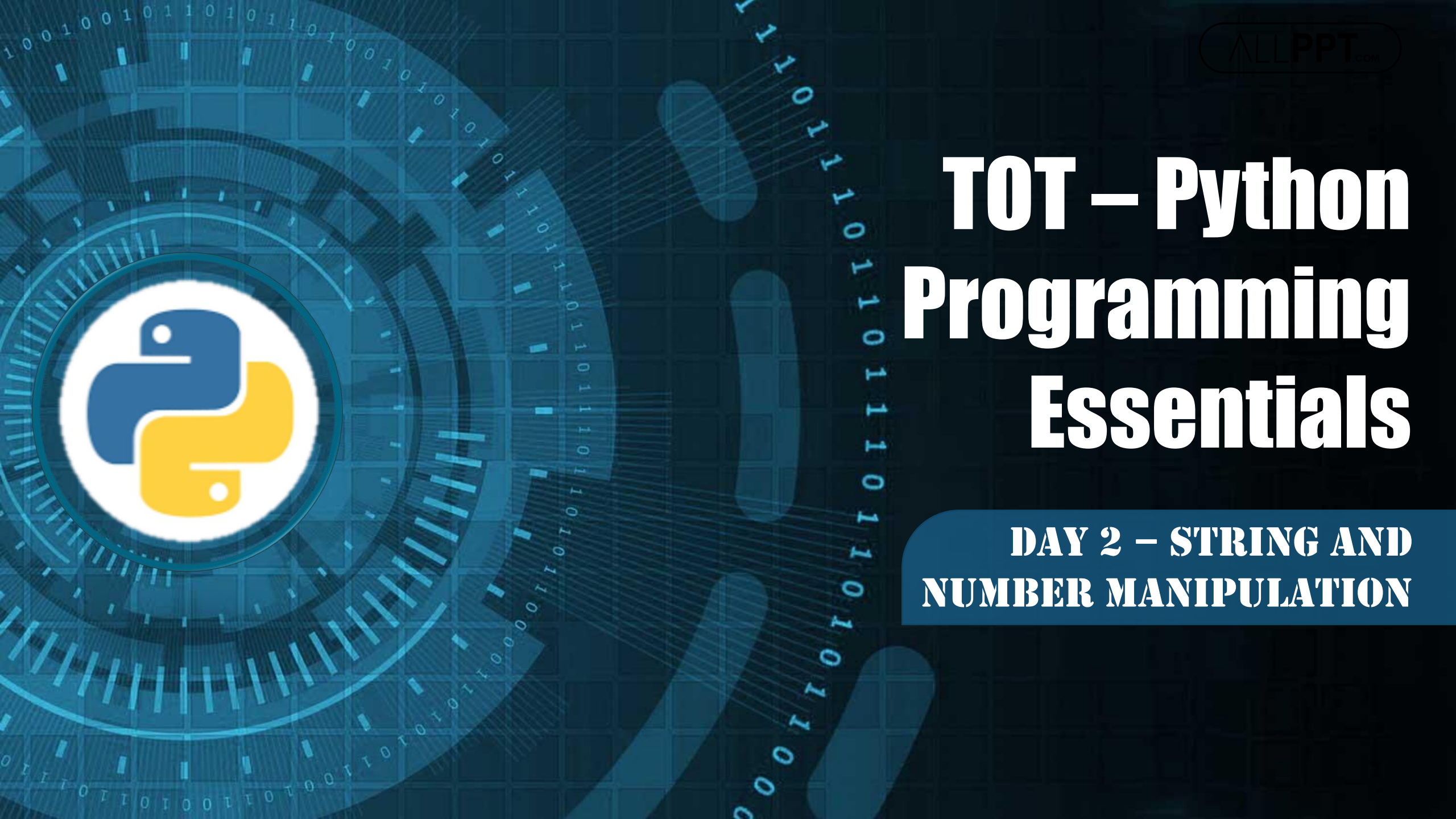

TOT – Python Programming Essentials

DAY 2 – STRING AND NUMBER MANIPULATION

TOPICS

01 Strings

02 Strings Placeholders

03 String Formatting Functions

04 Number Formatting Functions

05 Python Conditions

STRINGS

String – series of characters interpreted as text

```
"The quick brown fox. "
'The fast green turtle'
```

```
"This milk costs 30 pesos"
"Mike said 'I love eating burger!'"
```

```
'John said "I don\'t like vegetables"‘
"This next text will move to the \nnext line"
```

```
'd:\users\nathan'
r'd:\users\nathan'  # raw string
```

MULTI LINE STRINGS

You can assign a multiline string to a variable by using three quotes:

Double Quotes – """

```
lyrics = """I can think of all the times
You told me not to touch the light
I never thought that you would be the one
I couldn't really justify"""
print(lyrics)
```

Single Quotes – '''

```
Lyrics = '''I can think of all the times
You told me not to touch the light
I never thought that you would be the one
I couldn't really justify'''
print(lyrics)
```

STRINGS PLACEHOLDERS

Container for strings and numbers

%

Strings in Python have a unique built-in operation that can be accessed with the % operator. This lets you do simple positional formatting very easily.

{} format

The format() method formats the specified value(s) and insert them inside the string's placeholder.

f-Strings

f-strings lets you use embedded Python expressions inside string constants.

STRING PLACEHOLDERS {}

empty placeholders

```
sampleText1 = "My name is {} i love {} and playing {}"
sampleText1a = sampleText1.format(name, food, game)
print(sampleText1a)
```

numbered indexes

```
sampleText2 = "My name is {2} i love {1} and playing {0}"
sampleText2a = sampleText2.format(name, food, game)
print(sampleText2a)
```

named indexes

```
sampleText3 = "My name is {newname} i love {newfood} and playing {newgame}"
sampleText3a = sampleText3.format(newname="Mike", newfood="burger",
newgame="volleyball")
print(sampleText3a)
```

STRING PLACEHOLDERS %

%s and %f

```
item = “milk”
cost = 35.50
sampleText4 = "The product %s costs %.2f" % (item, cost)
print(sampleText4)
```

%c

```
print("The character after %c is %c." % ("B", "C"))
```

%i and %d

```
year = 2019
print("%i will be a perfect year." % year)
```

%10s and %-10s

```
place = "London"
print ("%10s is not a place in France" % place)  # Pad to the left
print ("%-10s is not a place in France" % place) # Pad to the right
```

STRING PLACEHOLDERS F-STRING AND {}

```
name = “Vinz”
sampleText5 = f“Hello sir {name}!"
print(sampleText5)
```

Inline arithmetic

```
a = 5
b = 10
>>> f'Five plus ten is {a + b} and not {2 * (a + b)}.'
```

STRING FORMATTING FUNCTIONS

upper()

Converts a string into upper case

***string*.upper()**

capitalize()

Converts the first character to uppercase

***string*.capitalize()**

split()

Splits the string at the specified separator, and returns a list

***string*.split(separator, maxsplit)**

lower()

Converts a string into lower case

***string*.lower()**

title()

Converts the first character of each word to uppercase

***string*.title()**

len()

Count characters in a string

len(*string*)

replace()

Returns a string where a specified value is replaced with a specified value

***string*.replace(oldvalue, newvalue, count)**

DAY 2
ACTIVITY 1

NUMBER FORMATTING FUNCTIONS

round()
returns a floating point number that is a rounded version of the specified number, with the specified number of decimals.
round(*number, digits*)

ceil()
rounds a number UP to the nearest integer, if necessary, and returns the result.
math.ceil(*x*)

floor()
rounds a number DOWN to the nearest integer, if necessary, and returns the result.
math.floor(*x*)

pow()
returns the value of x raised to power y.
math.pow(*x, y*)

EXAMPLES

```
x = round(5.76543)
print(x)
----------------------
x = round(5.76543, 2)
print(x)
```

```
import math

print(math.ceil(1.4))
print(math.ceil(5.3))
print(math.ceil(-5.3))
print(math.ceil(22.6))
print(math.ceil(10.0))
```

```
import math

print(math.floor(0.6))
print(math.floor(1.4))
print(math.floor(5.3))
print(math.floor(-5.3))
print(math.floor(22.6))
print(math.floor(10.0))

print(pow(2,3))
print(2**3)
```

DAY 2
ACTIVITY 2