PYTHON: WHAT-YOU-CAN-USE

Important:

- This is the list of instruction **you are allowed to use** when coding in PYTHON
- You cannot use any other PYTHON instruction because we want you to focus on your ALGORITHM
- We will update this list every week

	Instruction	Example
REPEATS	**for n in range(<NUMBER>) :**	**for n in range (3) :** print("hello") >hello >hello >hello
	for n in range(start, end)	**for n in range (2,5) :** print(n) >2 >3 >4
	for n in range(start, end, increment)	**for n in range (10,20,4) :** print(n) >10 >14 >18
	for student in students	vlues = [5, 6, 7, 8] For value in values: Print(value) >5 >6 >7 >8

	while <BOOLEAN> :	X = int(input())
		while X != 5:
		  print ("try again")
		  X = int(input())

CONDITION	IF :
	if <BOOLEAN> :
	  < instructions>
	
	IF/ELSE :
	if <BOOLEAN> :
	  < instructions>
	else :
	  < instructions>
		If x<5 and y>6 :
		  result = "monday"
		eiIf x> 10 :
		  result = "friday"
		else:
		  result = "sunday"

<table>
<tr><td></td><td>IF/ELSE IF :
if <BOOLEAN> :
< instructions>
esif <BOOLEAN> :
< instructions></td><td></td></tr>
<tr><td>INPUT</td><td>INPUT A STRING :

or
myString = input()

INPUT A INTEGER:
myInteger = int(input())

INPUT A FLOAT:
myFloat = float(input())</td><td>text = input()
print(text + “ hello”)

number = int(input())
number = number + 2</td></tr>
<tr><td>OUTPUT</td><td>OUTPUT WITH A BREAKLINE
print(<STRING>);

OUTPUT NO BREAKLINE
print(<STRING> , end=””)</td><td>Text = “Monday\nTuesday”
Print(text)

>Monday
>Tuesday</td></tr>
<tr><td>NUMBER OPERATORS</td><td>MATH OPERATIONS
<table><tr><td>ADD</td><td>MULTIPLY</td><td>DIVIDE</td><td>POWER</td></tr><tr><td>+</td><td>*</td><td>/</td><td>**</td></tr></table>
INCREMENT A VARIABLE VALUE:
x = x + 10

MODULO
x = y % 5

POWER
x = 3 ** 5</td><td>n1 = 4
n2 = n1 ** 2
print(n2)

>16

print(10%3)
>1
print(12%3)
>0

print(2** 4)
>16</td></tr>
<tr><td>STRING OPERATORS</td><td>CONCATENATE STRINGS
<STRING> + <STRING>

REMOVE THE LAST CHARACTERS:
<STRING> [: -1]

BREAK A LINE
text = “\n”

GET NUMBER OF CHARCTERS
count = len(<STRING>)</td><td>print(“ronan” + “hello”)
>ronanhello

print(“ronan”[:-2])
>ron

print(len(“ronan”))
>5

print(len(“hi\nho”))
>5</td></tr>
</table>

<table>
<tr>
<td></td>
<td>GET CHARACTER
char = text[3]

SUBSTRING
count = text[2:3] (start index + end index)
count = text[:3] (start index = 0 + end index)
count = text[2:] (start index + end index = number of chars)

CONVERT A STRING TO A NUMBER
number = int(<STRING>)

CHECK IF A STRING IS A NUMBER
booleanVariable= <STRING>.isNumeric()

CHANGE A STRING TO UPPERCASE
text = <STRING>.upper()</td>
<td>print(“abcd”[1])
>b

print(“abcde”[1:3])
>bc

print(int(“4”) + int(“5”))
>9

print(“4” + “5”)
>45

print(“hello”.isNumeric())
>False

print(“54”.isNumeric())
>True

print(“hello”.upper())
>HELLO</td>
</tr>
<tr>
<td>BOOLEAN OPERATORS</td>
<td>COMPARAISONS

<table><tr><td>EQUAL</td><td>DIFFERENT</td><td>GREATER</td><td>LOWER</td></tr><tr><td>==</td><td>!=</td><td>></td><td><</td></tr></table>
BOOLEAN OPERATIONS

<table><tr><td>AND</td><td>OR</td><td>NOT</td></tr><tr><td>and</td><td>or</td><td>!</td></tr></table></td>
<td>X = 5
print (x>2 and x<4 or x >=5)
>True</td>
</tr>
<tr>
<td>TYPE</td>
<td>CONVERT ANYTING OF STRING:
str(<ANYTHING>)

GET THE TYPE OF ANY VALUE:
type(<ANYTHING>)</td>
<td>str(5) -> “5”
str(False) -> “False”
str(“hello”) -> “hello”

type(5) -> class INT
type(5.1) -> class FLOAT
type(“5”) -> class STR
type(True) -> class BOOL</td>
</tr>
</table>

FUNCTION	DEFINE A FUNCTION
	def <**FUNCTION NAME**>(**PARAMETER**):
	<DO SOMETHING>
	<RETURN> <SOMETHING>
	
	CALL A FUNCTION (INVOKE)
	<FUNCTION NAME>(VALUE)
		`def sum(n1, n2):`
		`  total = n1 + n2`
		`  return total`
		
		`print(sum(100,200)) -> 300`

DATA
 	
STRUCTURES
	ARRAY
	# Create empty array
	array = []
	
	# Create array with values
	array = [12, 13, 15, 16]
	
	# Access using index
	value = array[2]
	
	# Insert value at index
	array.insert(1, 20)
	
	# Insert value at the end
	array.append(20)
	
	# Remove using index
	array.pop(2)
	
	# Get a sub array
	subarray = array[2:25]
	
	ARRAY 2D
	
	# Create array2D with values
	array2D = [[12, 13, 15, 16], [4, 5, 6, 7]]
	
	# Access using index
	value = array2D[2][0]
	
	DICTIONARY
	# Create empty dictionary
	dic = {}
	
	# Create array with values
	dic = { **key1**:**value1**, **key2**:**value2** … }
	
	# Access using **key**
	value = dic[**key1**]
	
	# Add value for a new key
	dic[**key3**] = **value3**
		`fruits = [“apple”, “banana”]`
		
		`fruits[1]`
		`> “banana”`
		
		`fruits.insert(0, “coconut”)`
		`> [“coconut”, “apple”, “banana”]`
		
		`fruits.append(“mango”)`
		`>  [“apple”, “banana”, “mango”]`
		
		`fruits.pop(1)`
		`> [“apple”]`
		
		`[“a”, “b”, “c”, “d”][1:3]`
		`> [“b”, “c”]`
		
		`[[1, 2, 3], [4, 5, 6]][0][2] = 3`
		
		`studentsAge = { }`
		`studentsAge['sokan'] = 25`
		
		`studentsAge['seiha'] = 95`
		`studentsAge[‘sokan'] = 35`
		
		`will produce :`
		`{`
		`  'sokan': 35,`

	# Update value from existing key dic[**key2**] = **value2New** # Remove using key dic. pop(**key2**)	‘seiha': 95, }