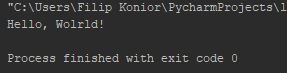
Sprawozdanie nr 2 Aplikacje Internetowe Filip Konior

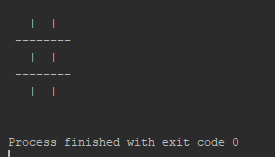
**1.Hello World**

def Hello():  
 print('Hello, Wolrld!')  
  
Hello()



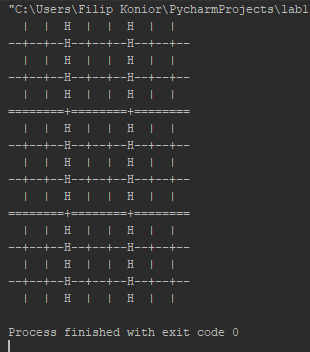
1. **Kółko i krzyżyk**

def tictactoeboard ():  
 a ="\n | | \n"  
 b ="--------"  
 print(a,b,a,b,a)  
  
tictactoeboard()



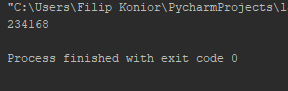
1. **Super kółko i krzyżyk**

def supertictactoeboard():  
 a = " | | H | | H | | "  
 b = "\n--+--+--H--+--+--H--+--+--\n"  
 c = "\n========+========+========\n"  
 print(" "+a, b,a,b,a,c,a,b,a,b,a,c,a,b,a,b,a)  
  
supertictactoeboard()



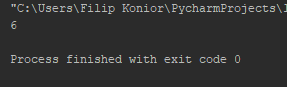
1. **Fizzbuzz (suma wielokrotności 3 albo 5)**

def fizzbuzz(n):  
 suma = 0  
 for i in range(n):  
 if i % 3 == 0 or i % 5 == 0:  
 suma += i  
 print(suma)



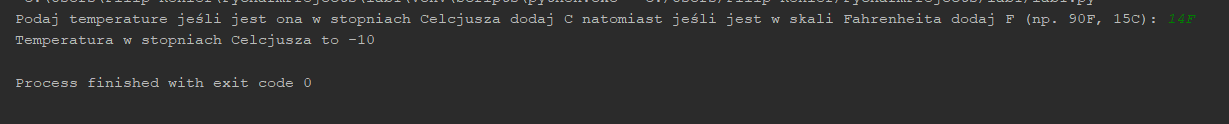
1. **Problem Collatza**

def Collatz(n):  
 dlugosc = 1  
 while n > 1:  
 if n % 2 == 0:  
 n /= 2  
 else:  
 n = 3 \* n + 1  
 dlugosc += 1  
 print(dlugosc)  
  
  
Collatz(5)



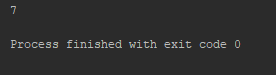
1. **Konwerter temperatur**

temp = input("Podaj temperature jeśli jest ona w stopniach Celcjusza dodaj C natomiast jeśli jest w skali Fahrenheita dodaj F (np. 90F, 15C): ")  
degree = int(temp[:-1])  
i\_convention = temp[-1]  
if i\_convention.upper() == "C":  
 result = int(round((9 \* degree) / 5 + 32))  
 o\_convention = "Fahrenheita"  
elif i\_convention.upper() == "F":  
 result = int(round((degree - 32) \* 5 / 9))  
 o\_convention = "Celcjusza"  
else:  
 print("źle wpisane dane.")  
 quit()  
print("Temperatura w stopniach", o\_convention, "to", result)



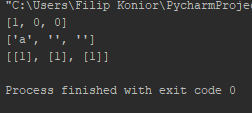
1. **Najwyższy wspólny dzielnik**

def gcd(a, b):  
 while b:  
 a, b = b, a % b  
 print (a)  
gcd(91,35)



1. **Listy**

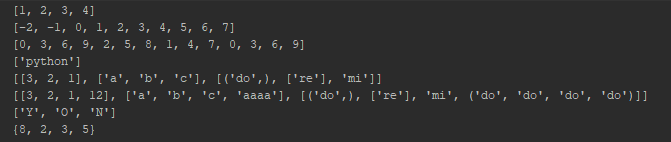
s = [0] \* 3  
s[0] += 1  
print(s) # [1, 0, 0]  
  
s = [''] \* 3  
s[0] += 'a'  
print(s) # ['a', '', '']  
  
s = [[]] \* 3  
s[0] += [1]  
print(s)



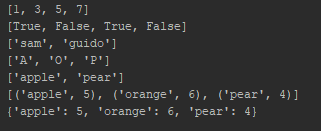
1. **Słowniki**

def flip\_dict(d):  
 out = {}  
 for key, value in d.items():  
 if value not in out:  
 out[value] = []  
 out[value].append(key)  
 return out

print([x for x in [1, 2, 3, 4]])  
print([n - 2 for n in range(10)])  
print([k % 10 for k in range(41) if k % 3 == 0])  
print([s.lower() for s in ['PythOn', 'iS', 'cOoL'] if s[0] < s[-1]])  
  
# Something is fishy here. Can you spot it?  
arr = [[3,2,1], ['a','b','c'], [('do',), ['re'], 'mi']]  
print(arr)  
[el.append(el[0] \* 4) for el in arr] # What does this return?  
print(arr)  
  
print([letter for letter in "pYthON" if letter.isupper()])  
print({len(w) for w in ["its", "the", "remix", "to", "ignition"]})



#[0, 1, 2, 3] -> [1, 3, 5, 7]  
print([x+i+1 for x,i in enumerate([0, 1, 2, 3])])  
  
#[3, 5, 9, 8] -> [True, False, True, False]  
print([k%3 == 0 for k in [3, 5, 9, 8]])  
  
#["TA\_sam", "TA\_guido", "student\_poohbear", "student\_htiek"] -> ["sam", "guido"]  
print([x[3:] for x in ["TA\_sam", "TA\_guido", "student\_poohbear", "student\_htiek"] if x.startswith("TA\_")])  
  
#['apple', 'orange', 'pear'] -> ['A', 'O', 'P']  
print([x[0].upper() for x in ['apple', 'orange', 'pear']])  
  
#['apple', 'orange', 'pear'] -> ['apple', 'pear']  
print([x for x in ['apple', 'orange', 'pear'] if len(x) < 6])  
  
#['apple', 'orange', 'pear'] -> [('apple', 5), ('orange', 6), ('pear', 4)]  
print([(x,len(x)) for x in ['apple', 'orange', 'pear']])  
  
#['apple', 'orange', 'pear'] -> {'apple': 5, 'orange': 6, 'pear': 4}  
d = {}  
[d.update({x:len(x)}) for x in ['apple', 'orange', 'pear']]  
print(d)



import math  
  
  
def is\_cyclone\_phrase(phrase):  
 for p in phrase.split():  
 m=int(math.ceil(len(p)/2))  
 if len(p) == 0 or len(p) == 1:  
 break  
 else:  
 if len(p) == 2:  
 if p[0] > p[1]:  
 return False  
 else:  
 for n,i in enumerate(p):  
 if n + 1 == m:  
 break  
 if p[n] > p[-1-n]:  
 return False  
 if p[-1-n] >p[n+1]:  
 return False  
 return True  
  
print(is\_cyclone\_phrase("adjourned")) # => True  
print(is\_cyclone\_phrase("settled")) # => False  
print(is\_cyclone\_phrase("all alone at noon")) # => True  
print(is\_cyclone\_phrase("by myself at twelve pm")) # => False  
print(is\_cyclone\_phrase("acb")) # => True  
print(is\_cyclone\_phrase("")) # => True  
  
print(is\_cyclone\_phrase("abcca")) # => True  
print(is\_cyclone\_phrase("accdb")) # => False

