

Нw1:

1) Составить таблицу соответствия между различными объектами основных классов: int, str и объектами класса bool.

int	bool
1	True
0	False
-1	True
9	True
-9	True

str	bool
""	False
"1"	True
"abc"	True
"-9sdf"	True
"0"	True

2) Разобраться с различиями между input() и raw_input(). Также в контексте разных версий python: 2 и 3.

В Python2:

input() сама определяет тип вводимых данных. Пример:

1) >>> num=input("Enter number: ")

Enter number: 'a'

>>> num

'a'

2) >>> num=input("Enter number: ")

Enter number: 5

>>> num

5

raw_input() считывает данные и оставляет их тип как str. Пример:

1) >>> num=raw_input("Enter number: ")

Enter number: a

>>> num

'a'

2) >>> num=raw_input("Enter number: ")

Enter number: 5

>>> num

'5'

В Python3:

input() воспринимает всё как строку. Пример:

1) >>> num=input("Enter number: ")

```
Enter number: a
>>> num
'a'
2) >>> num=input("Enter number: ")
Enter number: 5
>>> num
'5'
A raw_input() в Python3 нет.
```

3) Найти и прочитать PEP про изменение print между python2 и python3.

PEP 3105 -- Make print a function:
Описание новой (в Python3) функции print():
def print(*args, sep=' ', end='\n', file=None)

Python3: print(a, b, c, file=sys.stderr)
Python2: print >>sys.stderr, a, b, c

Необязательные аргументы sep и end указывают, что печатается между и после аргументов соответственно.

Характерная черта, которая называется softspace, т.е. полускрытый атрибут, используемый для указания “printy” в Python2, следует ли вставлять пробел перед первым элементом, была удалена. Поэтому прямого преобразования следующих выражений нет: print “a”, print , который не будет печатать пробел между "a" и новой строкой.

Обратная совместимость: print в Python2 при переносе в Python3 не будет работать, только если случайно не были поставлены скобки и из них будут продолжать делать то же самое только те, которые печатают одно заключенное в скобки значение.

Например в Python2:

```
>>> print ("Hello")
Hello
>>> print ("Hello", "world")
('Hello', 'world')
```

А в Python3:

```
>>> print ("Hello")
Hello
>>> print ("Hello", "world")
Hello world
```

Всё же, т.к. print - это оператор в Python 2, печать может быть обнаружена и заменена надёжно и без двусмысленности автоматизированным инструментом.