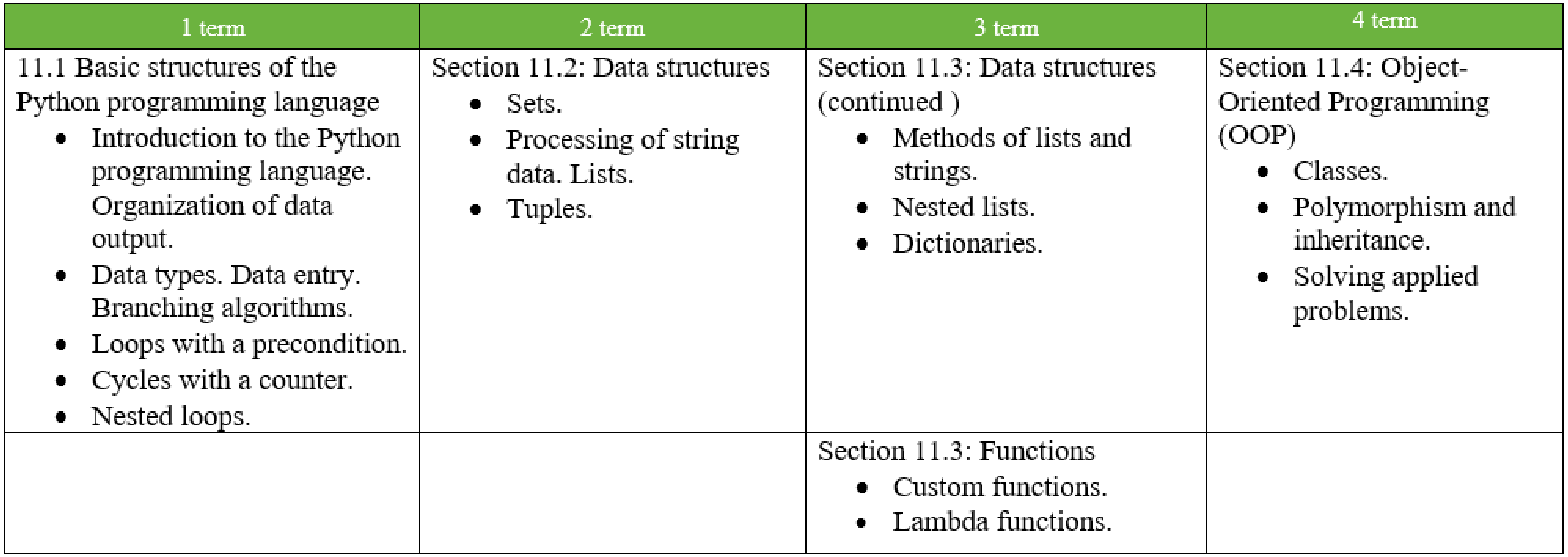
Programming

11 grade

Long term plan



1 term	2 term	3 term	4 term
11.1 Basic structures of the Python programming language • Introduction to the Python programming language. Organization of data output. • Data types. Data entry. Branching algorithms. • Loops with a precondition. • Cycles with a counter. • Nested loops.	Section 11.2: Data structures • Sets. • Processing of string data. Lists. • Tuples.	Section 11.3: Data structures (continued) • Methods of lists and strings. • Nested lists. • Dictionaries.	Section 11.4: Object-Oriented Programming (OOP) • Classes. • Polymorphism and inheritance. • Solving applied problems.
		Section 11.3: Functions • Custom functions. • Lambda functions.	

Assessment

SAU = 15 points

- 6 points for SAU
- + 9 points for 1 lesson

1 Lesson

- 2-4 tasks for beginners
- + tasks for advanced students

Introduction to Python. Output data.

11.1.1.1 organise data output;

11.1.1.2 use the escape sequences with data output.

What do you know about Python?

Python is a popular programming language.

It was created by Guido van Rossum, and released in 1991.

The most recent major version of Python is Python 3. However, Python 2, is still quite popular.

A python file has **.py** extension

What can Python do?

- Python can be used on a server to create web applications.
- Python can be used alongside software to create workflows.
- Python can connect to database systems. It can also read and modify files.
- Python can be used to handle big data and perform complex mathematics.
- Python can be used for rapid prototyping, or for production-ready software development.

Why Python?

- Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
- Python has a simple syntax similar to the English language.
- Python has syntax that allows developers to write programs with fewer lines than some other programming languages.
- Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.
- Python can be treated in a procedural way, an object-oriented way or a functional way.

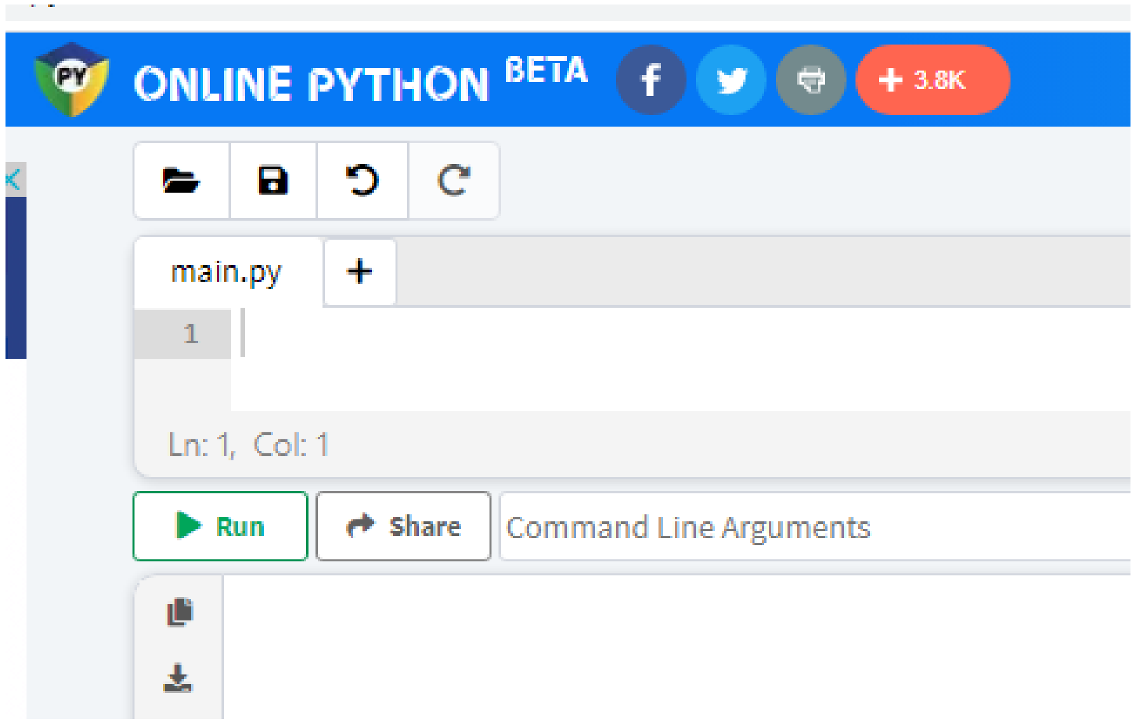
https://www.youtube.com/watch?v=G3vu8RCrZEc

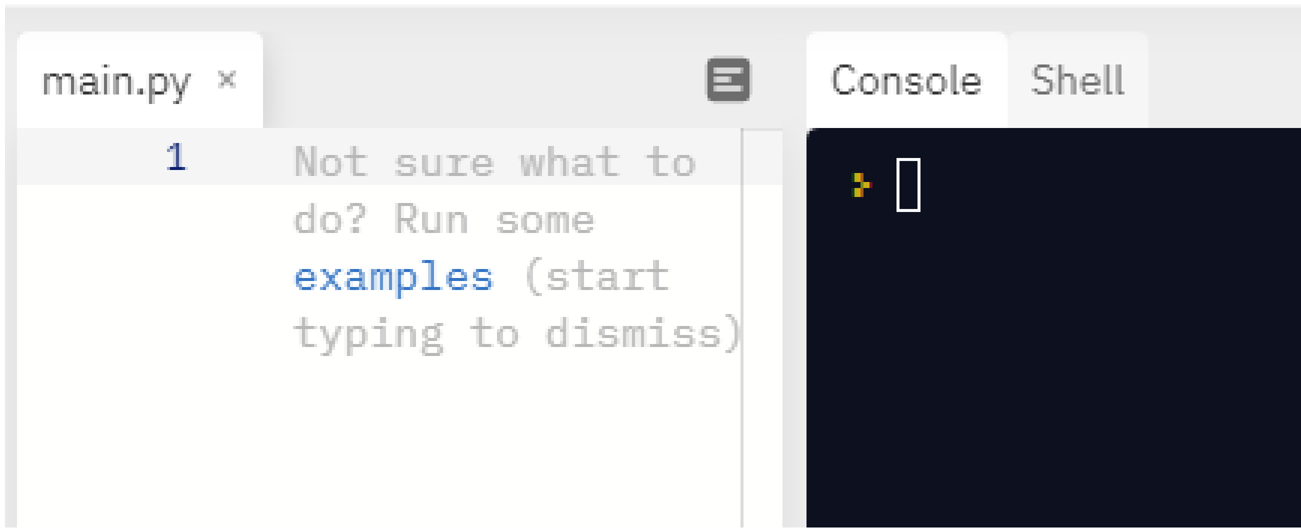
Python Syntax compared to other programming languages

- Python **was designed for readability**, and has some similarities to the English language with influence from mathematics.
- Python **uses new lines to complete a command**, as opposed to other programming languages which often use semicolons or parentheses.
- Python **relies on indentation**, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose.

Online compilers/interpreters

https://www.online-python.com/ https://replit.com



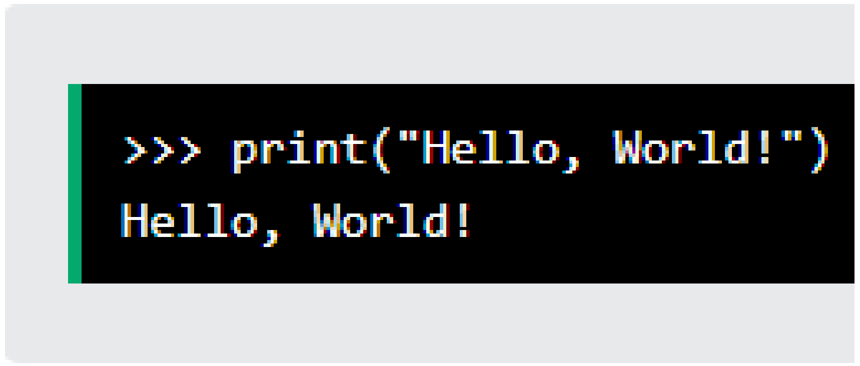


```
Not sure what to
do? Run some
examples (start
typing to dismiss)
```

11.1.1.1 organize data output;

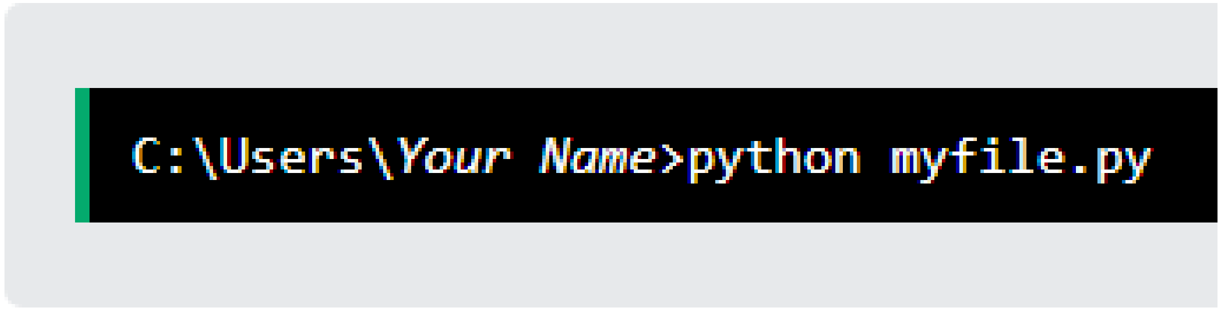
Execute Python Syntax

- Python syntax can be executed by writing directly in the Command Line:



```
>>> print("Hello, World!")
Hello, World!
```

- Or by creating a python file on the server, using the .py file extension, and running it in the Command Line:
- C:\Users\Your Name>python myfile.py



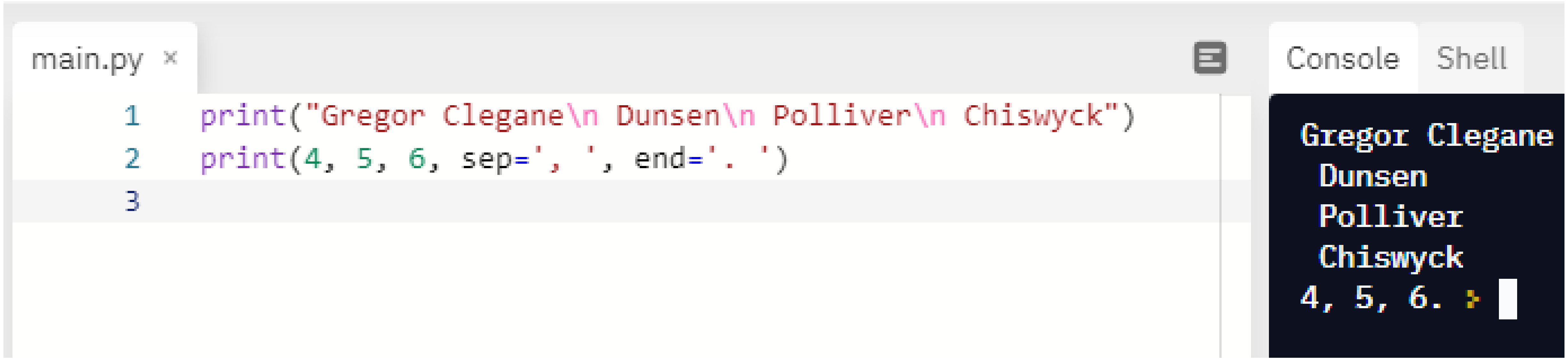
```
C:\Users\Your Name>python myfile.py
```

Escaped sequences

Escaped sequences are sequences that start with the character **"\\"** followed by one or more characters.

For example:

"\n" - line break



main.py

```python
print("Gregor Clegane\n Dunsen\n Polliver\n Chiswyck")
print(4, 5, 6, sep=', ', end='. ')

```

Console Shell

```
Gregor Clegane
 Dunsen
 Polliver
 Chiswyck
4, 5, 6. 
```

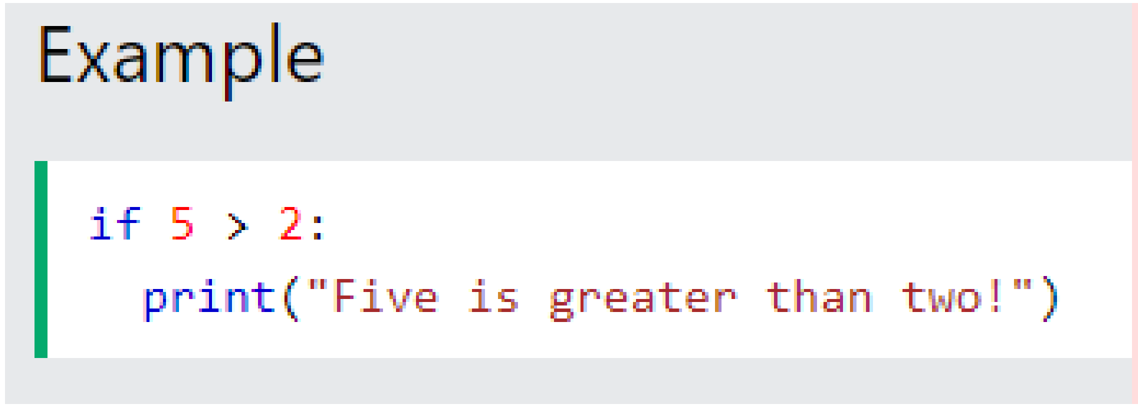
Escaped sequences

\\	Allows you to write a backslash character.
\’	Allows you to write a single apostrophe character
\”	Allows you to write a single quote character.
\n	Line break (new line).
\r	Returns the cursor to the beginning of the line.
\t	Horizontal indentation to the left of the beginning of the line (horizontal tab).
\v	Vertical indentation at the top (vertical tab).

https://pyprog.pro/python/py/str/esqape_sec.html

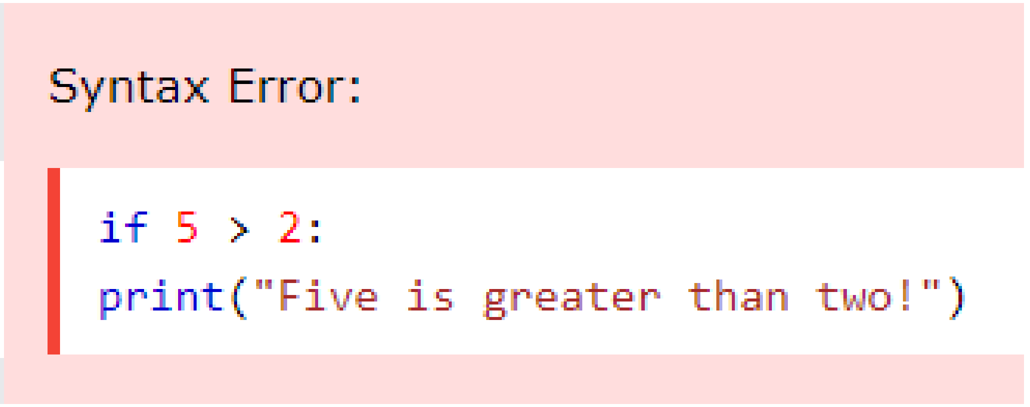
Python Indentation

- Indentation refers to the spaces at the beginning of a code line.
- Where in other programming languages the indentation in code is for readability only, **the indentation in Python is very important.**
- Python uses **indentation to indicate a block of code.**



Example

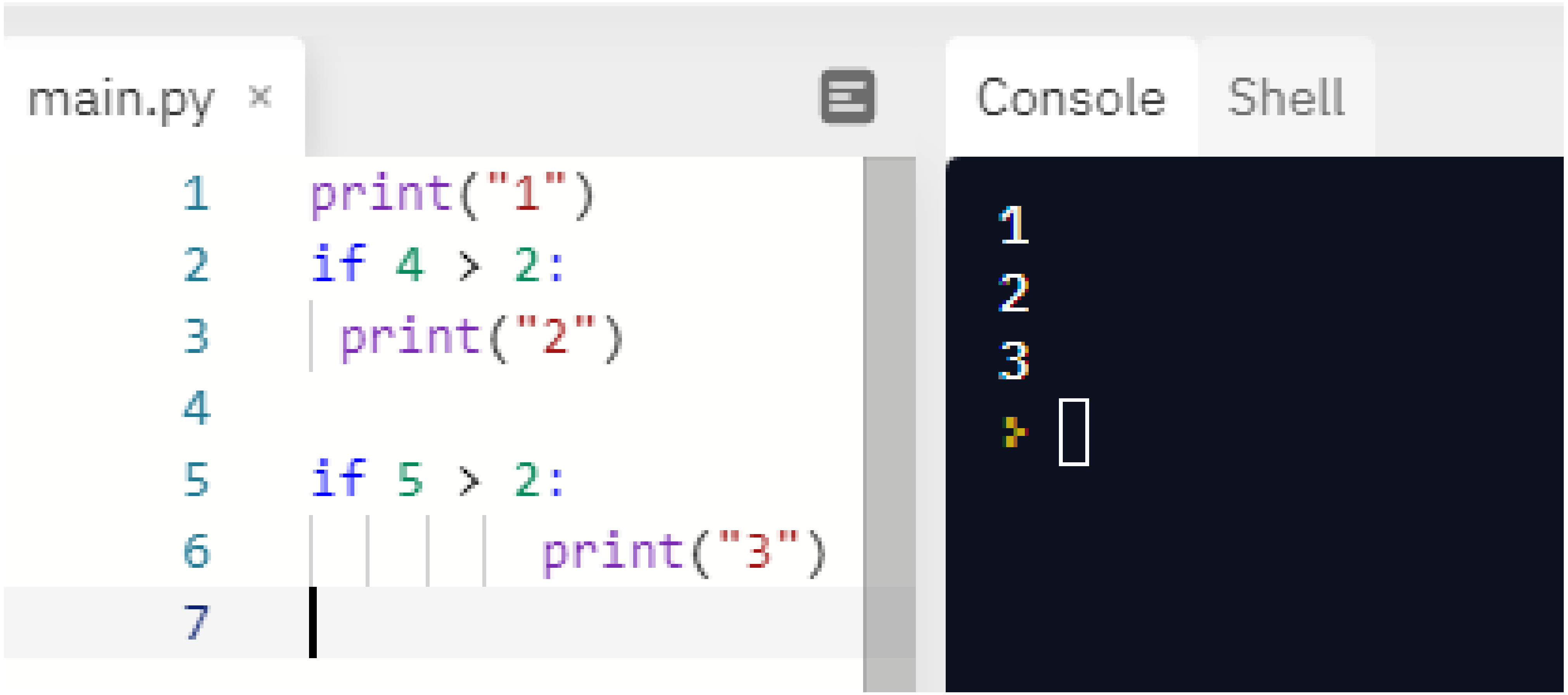
```python
if 5 > 2:
  print("Five is greater than two!")
```



Syntax Error:

```python
if 5 > 2:
print("Five is greater than two!")
```

The number of spaces is up to you as a programmer, but it has to be at least one!!!



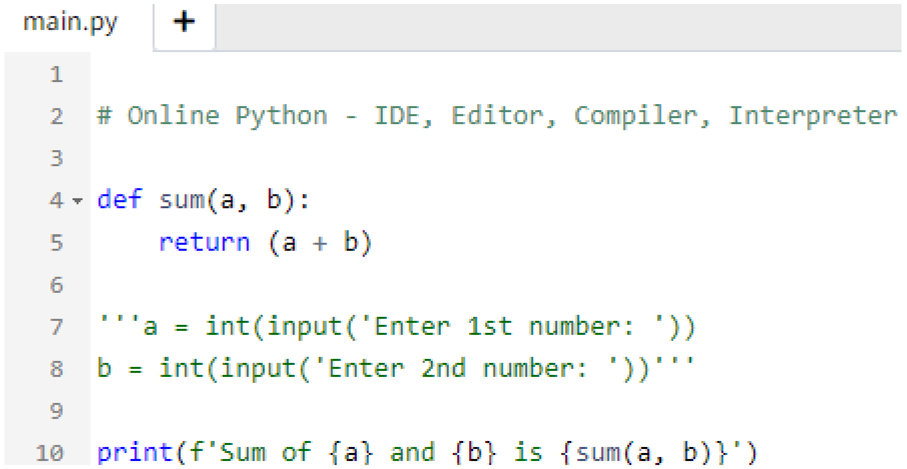
```python
print("1")
if 4 > 2:
 print("2")

if 5 > 2:
        print("3")
```

```
1
2
3
> 
```

Comments

- Python has commenting capability for the purpose of in-code documentation.
- Comments start with a #, and Python will render the rest of the line as a comment:
- To comment out multiple lines in Python, you can use triple quotes (''' ''' or """ """) for block comments



main.py +

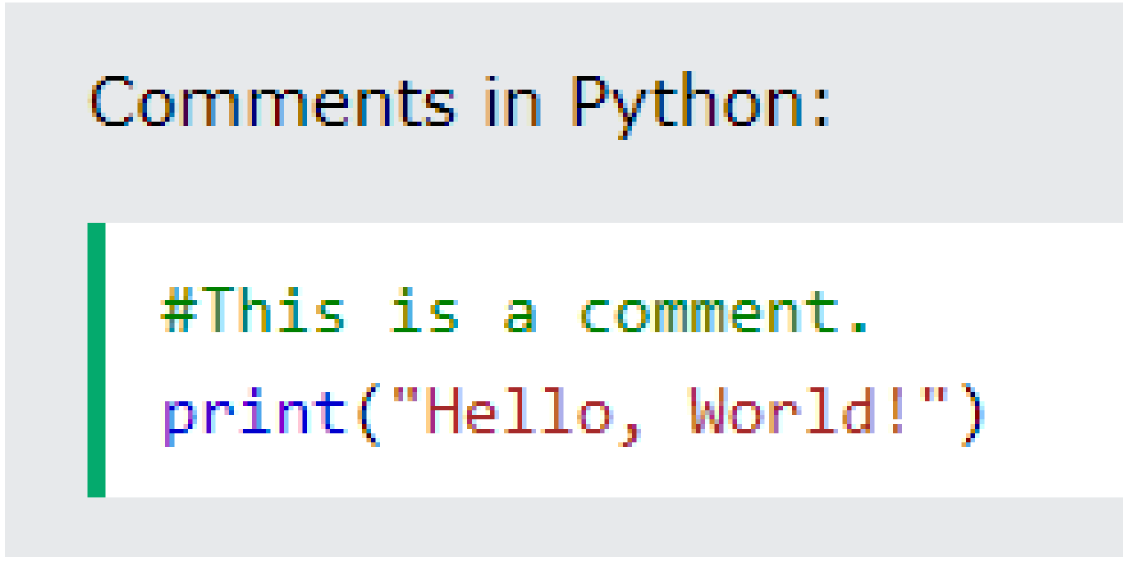
```python

# Online Python - IDE, Editor, Compiler, Interpreter

def sum(a, b):
    return (a + b)

'''a = int(input('Enter 1st number: '))
b = int(input('Enter 2nd number: '))'''

print(f'Sum of {a} and {b} is {sum(a, b)}')
```



Comments in Python:

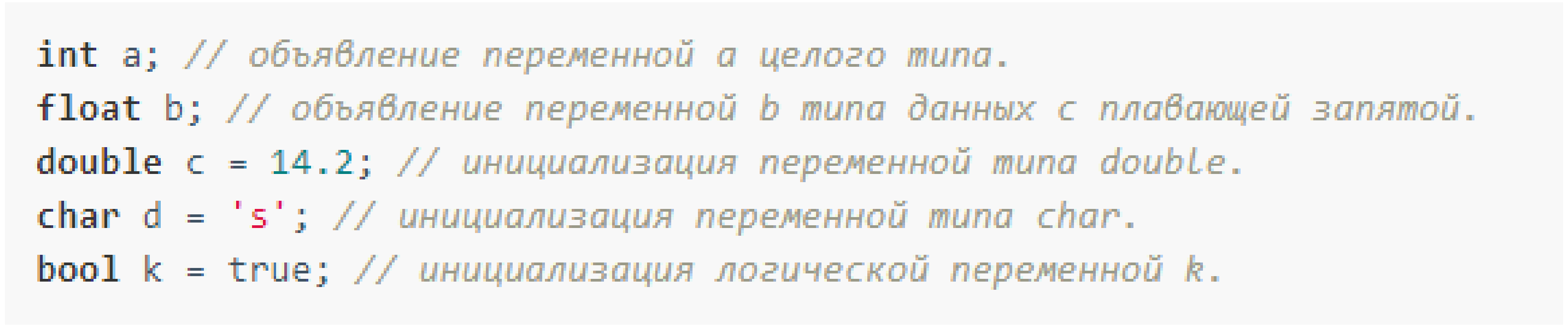
```python
#This is a comment.
print("Hello, World!")
```

What is a variable?

Variables are containers for storing data values.

How variables are declared in C++?

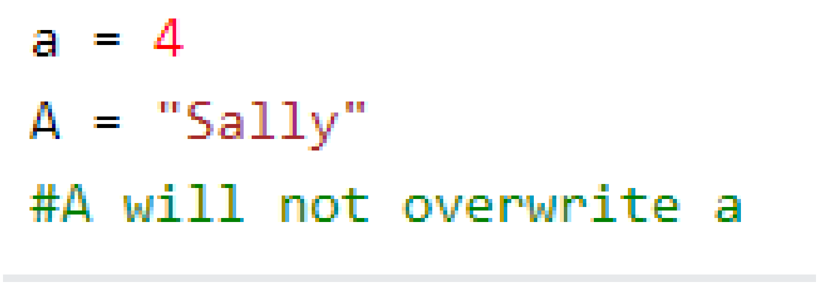
What is a variable type?



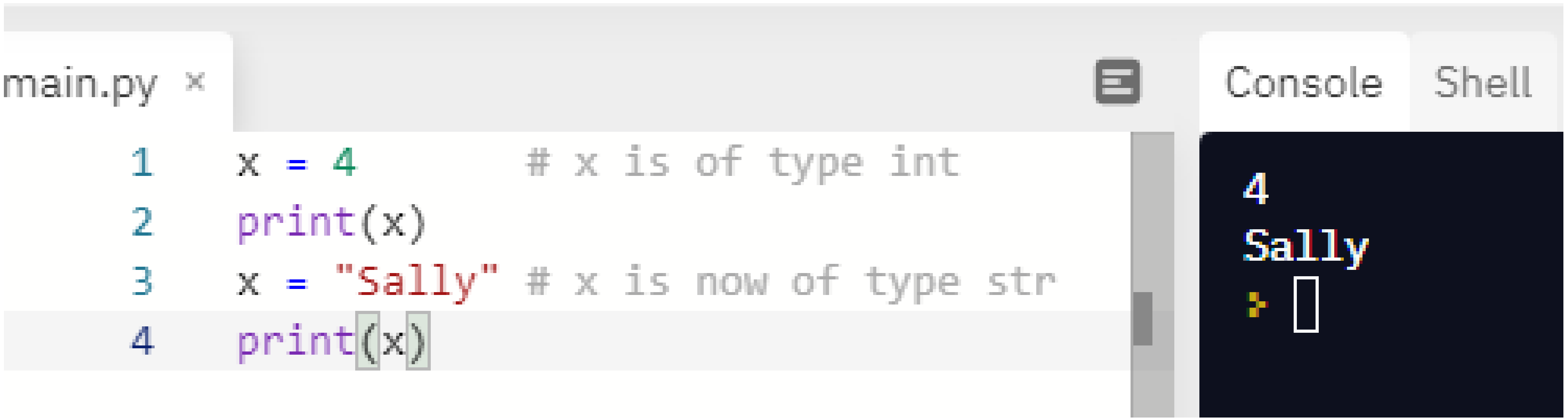
```cpp
int a; // объявление переменной a целого типа.
float b; // объявление переменной b типа данных с плавающей запятой.
double c = 14.2; // инициализация переменной типа double.
char d = 's'; // инициализация переменной типа char.
bool k = true; // инициализация логической переменной k.
```

Python Variables

- Python has no command for declaring a variable.
- Variables do not need to be declared with any particular *type*
- They can change type after they have been set.
- Variable names are case-sensitive.



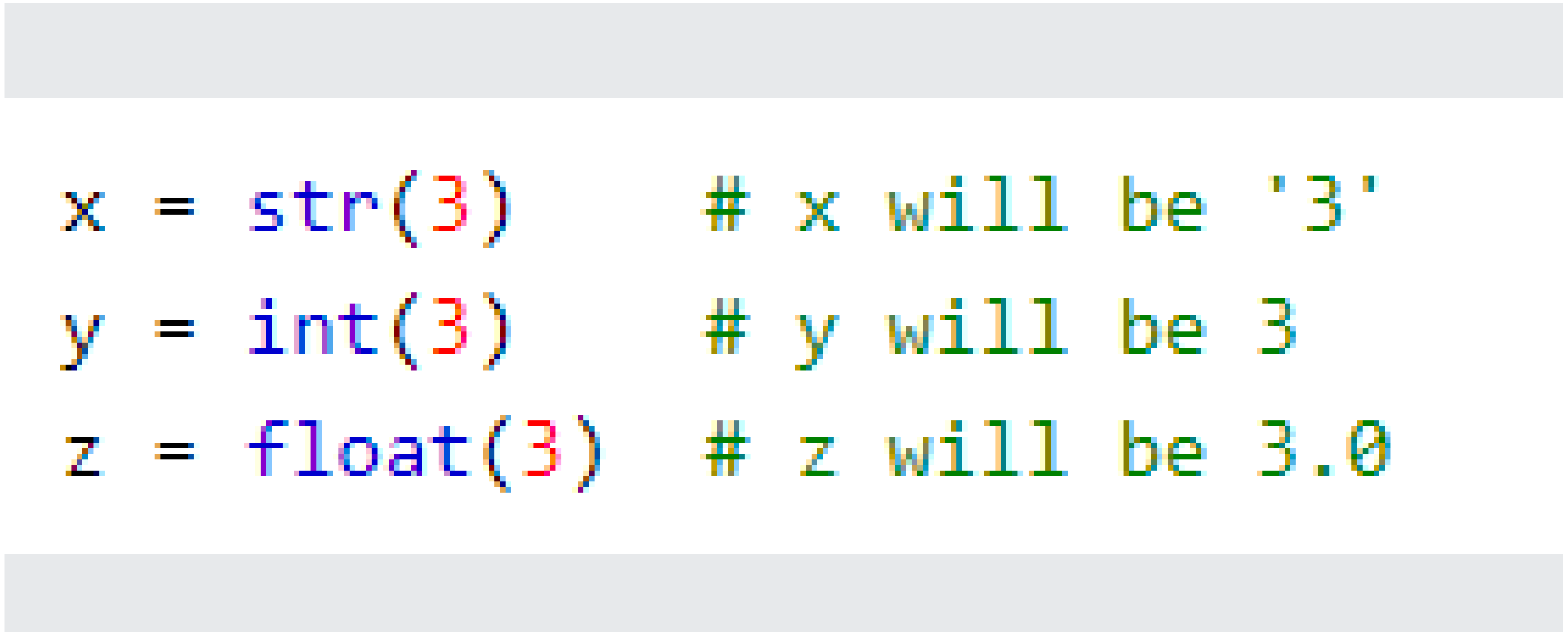
```
a = 4
A = "Sally"
#A will not overwrite a
```



```
x = 4          # x is of type int
print(x)
x = "Sally" # x is now of type str
print(x)
```

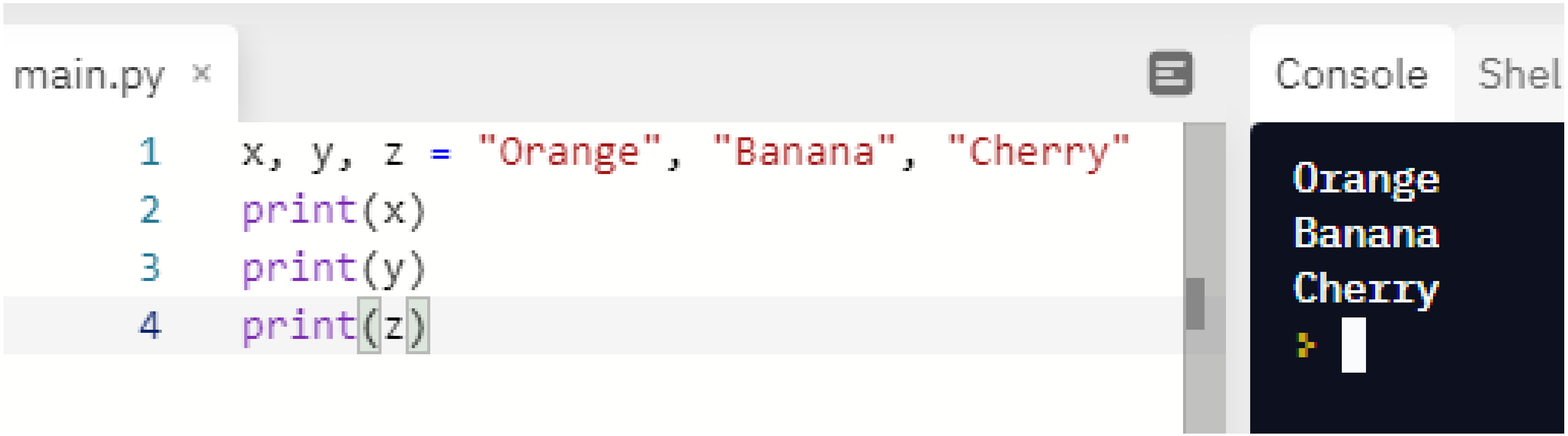
```
4
Sally
```

But if you want to specify the data type of a variable, this can be done with casting.



```
x = str(3)    # x will be '3'
y = int(3)    # y will be 3
z = float(3)  # z will be 3.0
```

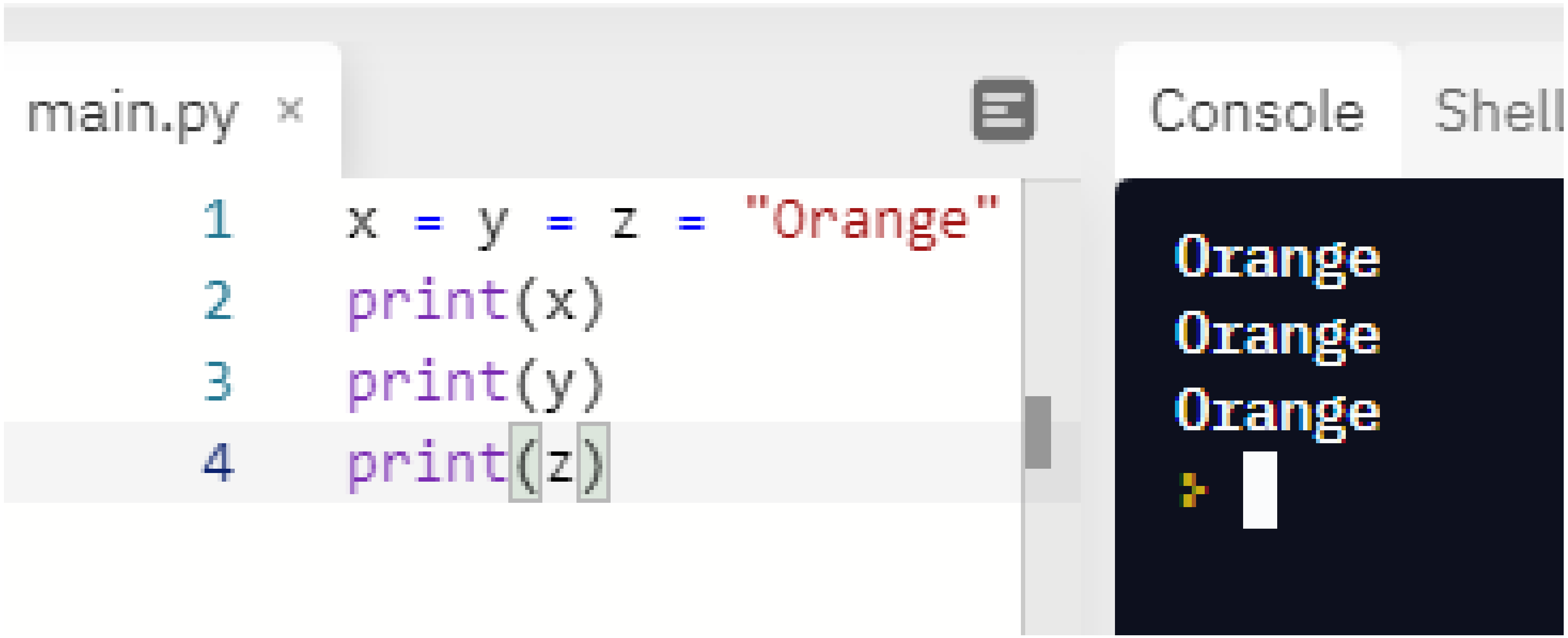
Many Values to Multiple Variables



```python
x, y, z = "Orange", "Banana", "Cherry"
print(x)
print(y)
print(z)
```

```
Orange
Banana
Cherry
›
```

One Value to Multiple Variables



main.py ×

```python
x = y = z = "Orange"
print(x)
print(y)
print(z)
```

```
Orange
Orange
Orange
> 
```

What is a data type?

What the data types do you know?

Data types. Input data.

11.1.1.3 distinguish between data types in Python;

11.1.1.4 convert data types of variables;

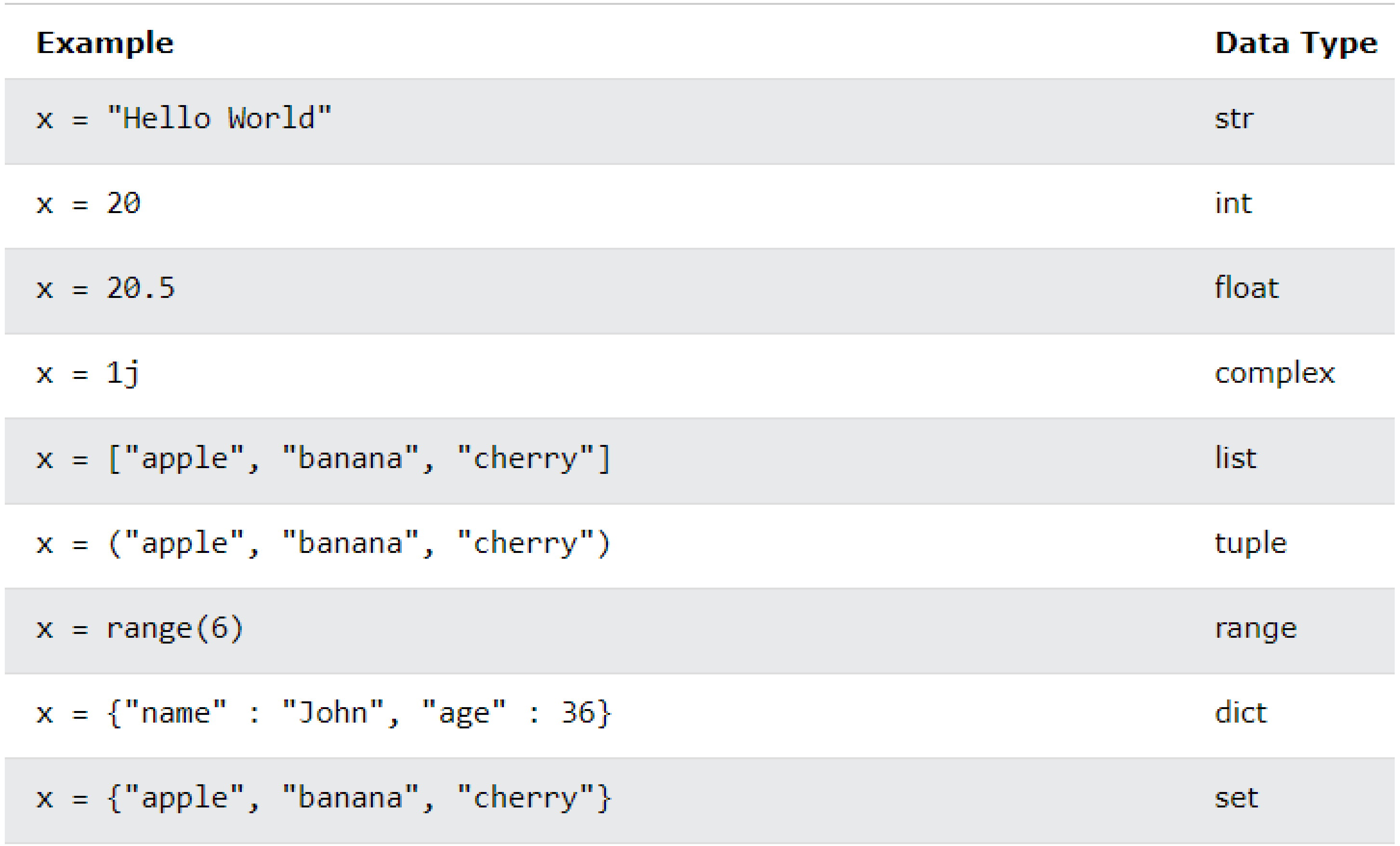
11.1.1.5 organize keyboard inputs;

11.1.1.6 use arithmetic operations when solving problems;

11.4.3.2 solve applied problems of various subject areas.

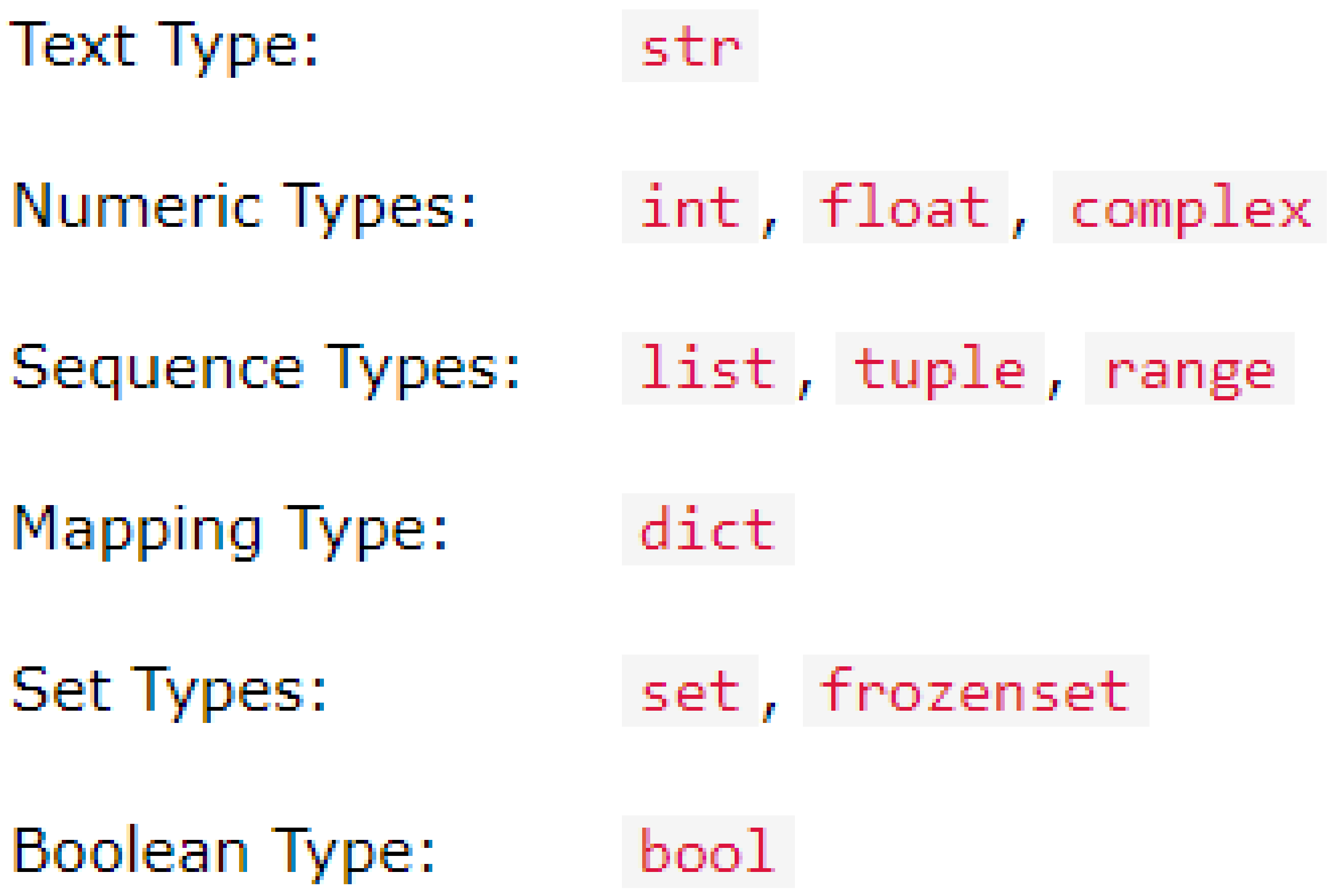
Data Type

- Variables can store data of different types, and different types can do different things.
- You can get the data type of any object by using the type() function
- In Python, the data type is set when you assign a value to a variable



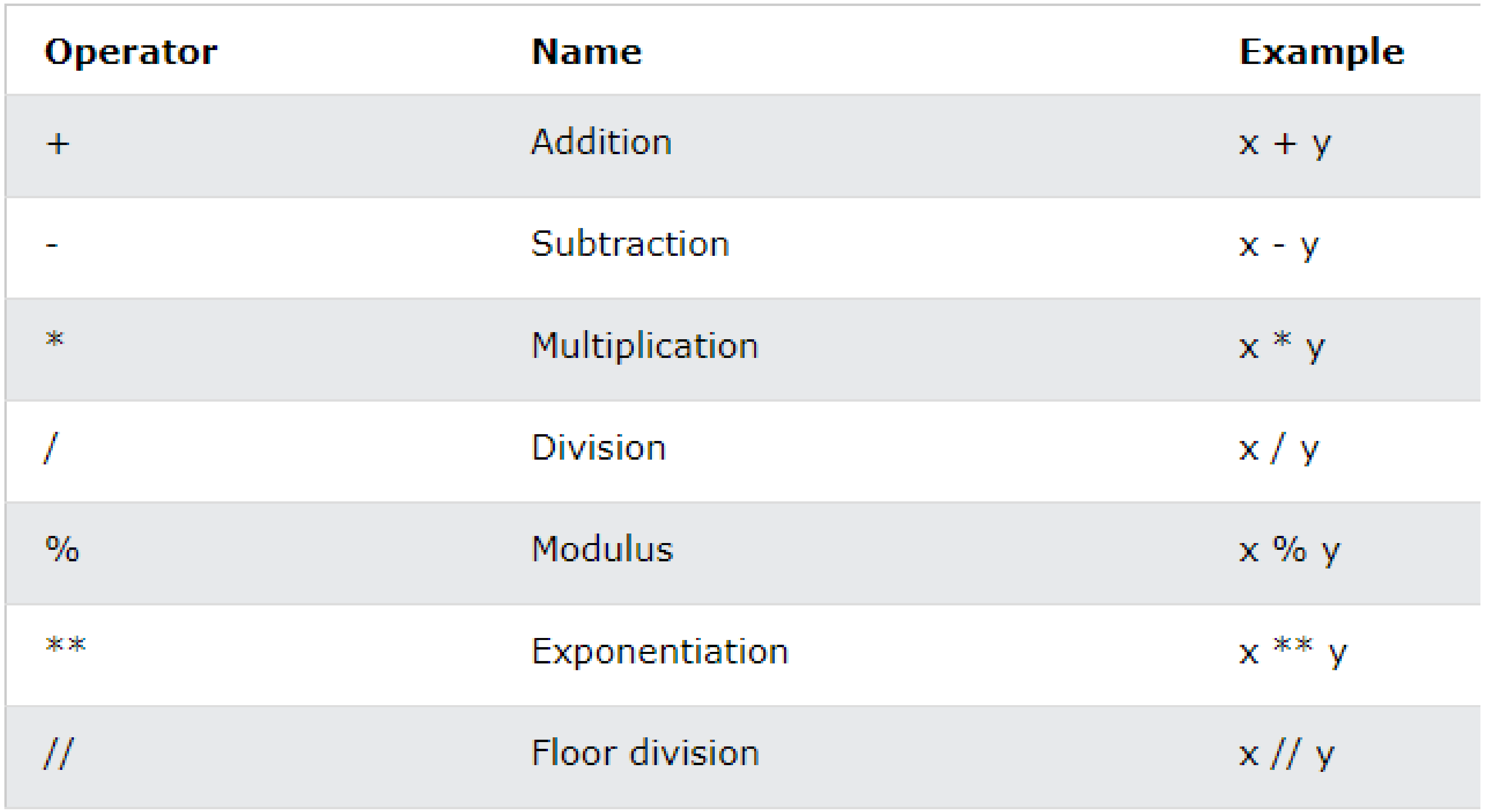
Example	Data Type
`x = "Hello World"`	str
`x = 20`	int
`x = 20.5`	float
`x = 1j`	complex
`x = ["apple", "banana", "cherry"]`	list
`x = ("apple", "banana", "cherry")`	tuple
`x = range(6)`	range
`x = {"name" : "John", "age" : 36}`	dict
`x = {"apple", "banana", "cherry"}`	set

Built-in Data Types



Text Type:	`str`
Numeric Types:	`int`, `float`, `complex`
Sequence Types:	`list`, `tuple`, `range`
Mapping Type:	`dict`
Set Types:	`set`, `frozenset`
Boolean Type:	`bool`

Python Arithmetic Operators

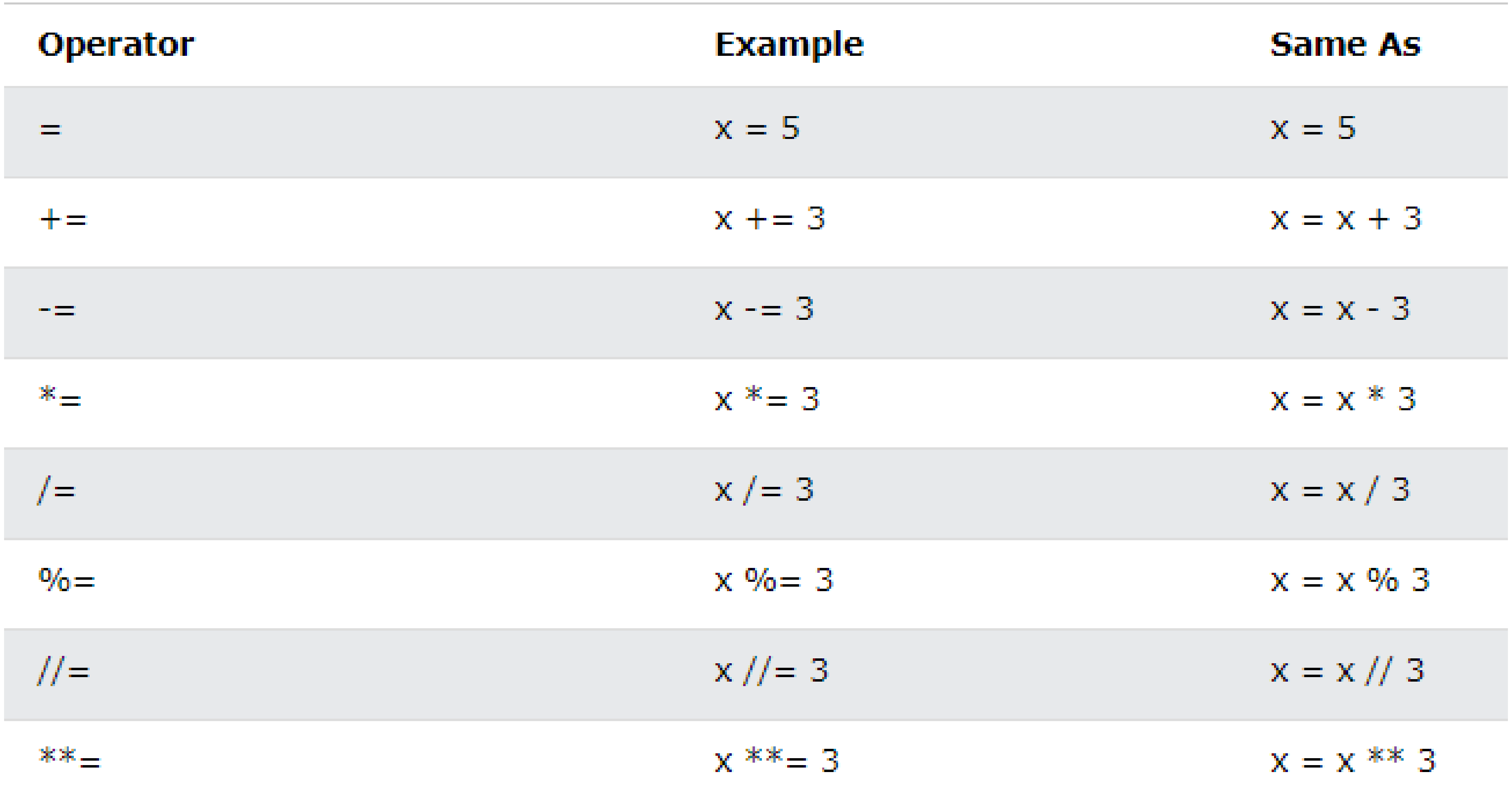


Operator	Name	Example
+	Addition	x + y
-	Subtraction	x - y
*	Multiplication	x * y
/	Division	x / y
%	Modulus	x % y
**	Exponentiation	x ** y
//	Floor division	x // y

Calculate the result of the operation

• 5/2	• 2,5
• 10%3	• 1
• 6%1	• 0
• 7%10	• 7
• 2**3	• 8
• 3**2	• 9
• 9//2	• 4
• -9//2	• -5

Python Assignment Operators



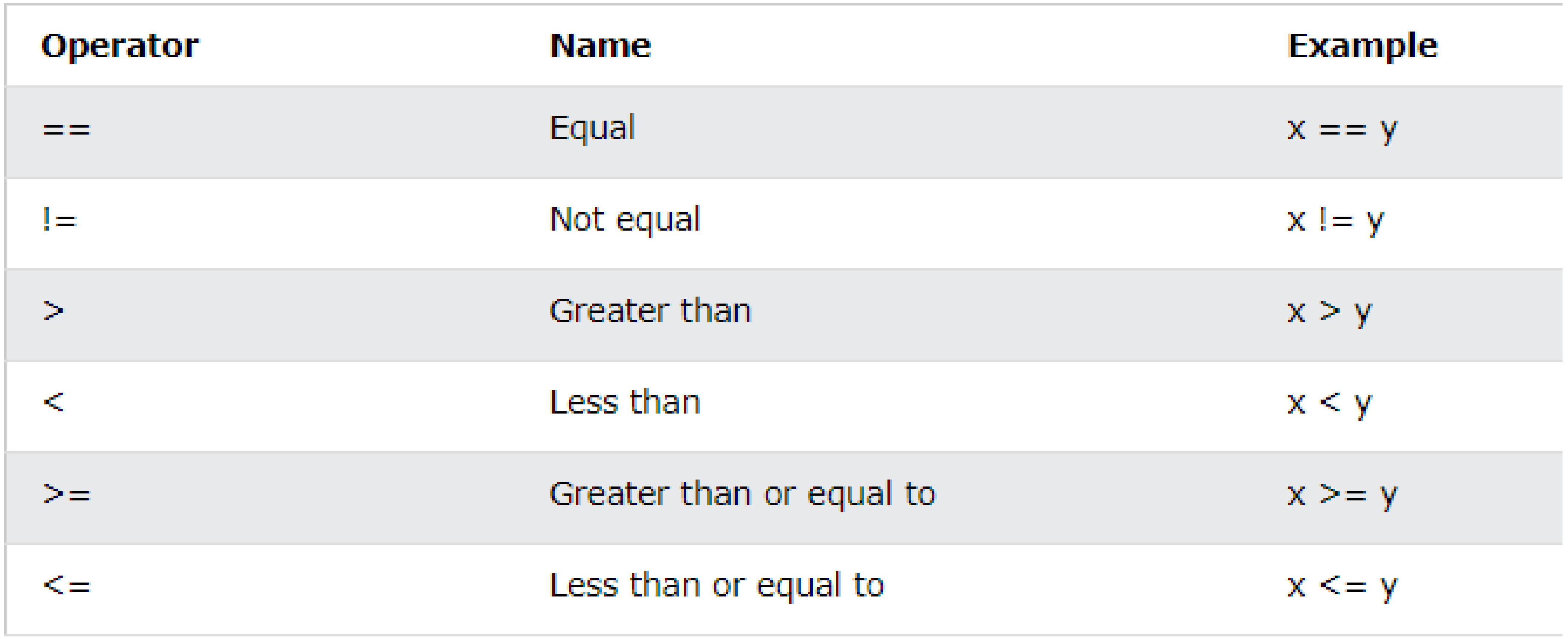
Operator	Example	Same As
=	x = 5	x = 5
+=	x += 3	x = x + 3
-=	x -= 3	x = x - 3
*=	x *= 3	x = x * 3
/=	x /= 3	x = x / 3
%=	x %= 3	x = x % 3
//=	x //= 3	x = x // 3
**=	x **= 3	x = x ** 3

Calculate the result of the operation
x=your day of birth
y= your month of birth

1 task
0,5 point

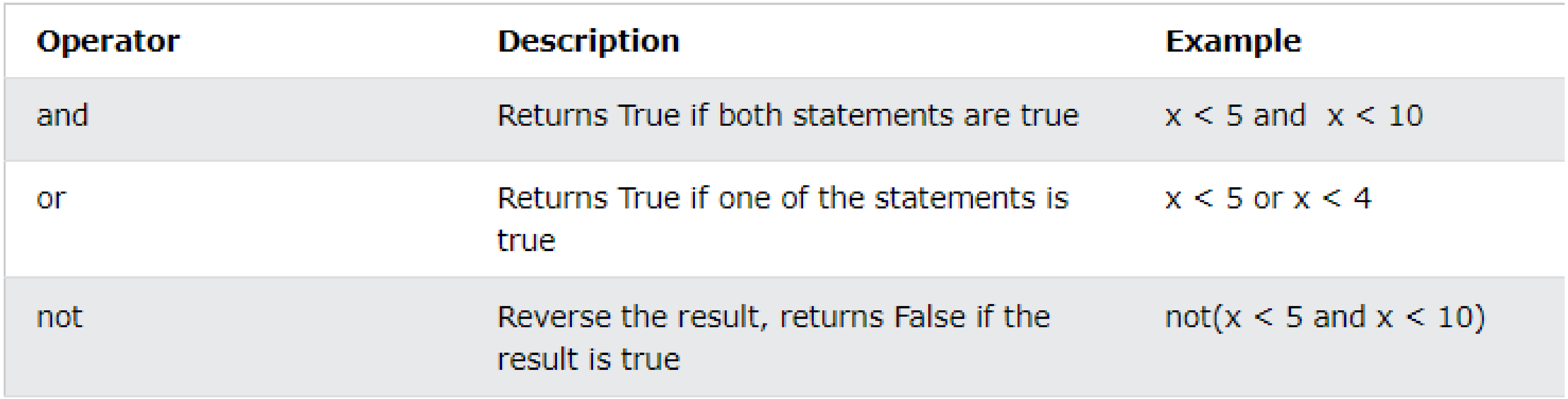
- X+=Y
- X-=Y
- X*=2
- X/=2
- X//=3
- X**=2
- X%=Y

Python Comparison Operators



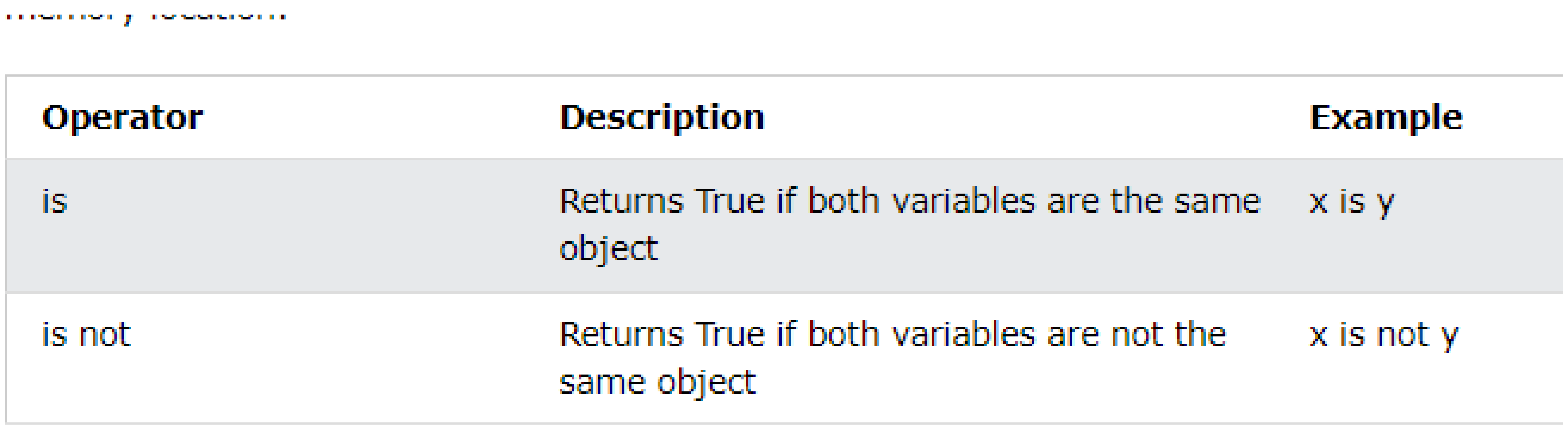
Operator	Name	Example
==	Equal	x == y
!=	Not equal	x != y
>	Greater than	x > y
<	Less than	x < y
>=	Greater than or equal to	x >= y
<=	Less than or equal to	x <= y

Python Logical Operators



Operator	Description	Example
and	Returns True if both statements are true	x < 5 and x < 10
or	Returns True if one of the statements is true	x < 5 or x < 4
not	Reverse the result, returns False if the result is true	not(x < 5 and x < 10)

Python Identity Operator



memory location.

Operator	Description	Example
is	Returns True if both variables are the same object	x is y
is not	Returns True if both variables are not the same object	x is not y

Calculate the result of the operation
x=5
y=3

a) X<3 or y>2	a) True
b) X==5 and y!=3	b) False
c) Not(x>=7)	c) True
d) X is Y	d) False
e) X is (y+2)	e) True
f) X%2 is not y	f) true

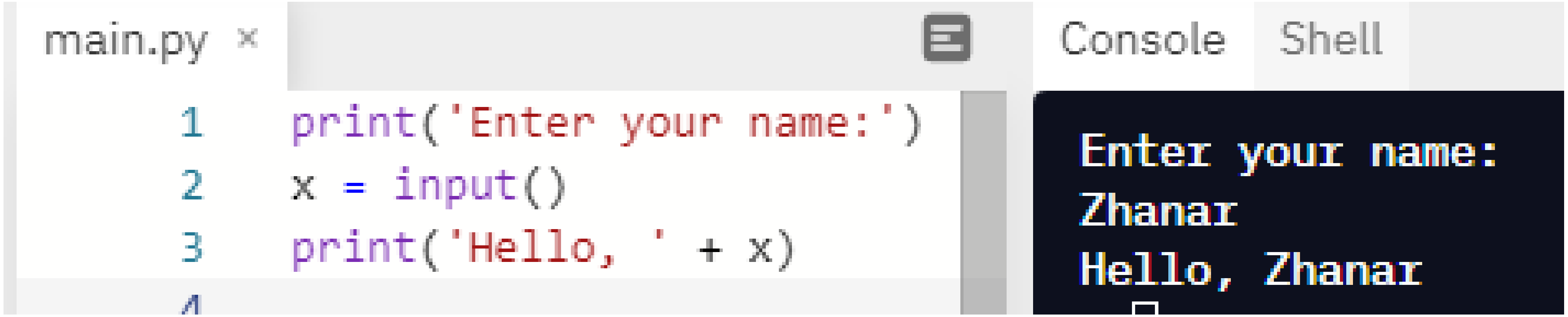
Calculate the result of the operation x=your day of birth y= your month of birth

2 task 0,5 point

a) X>12 and y>5
b) X==5 or y!=3
c) Not(x<=7)
d) X is not Y
e) X%2 is y

2 LESSON

Python input() Function



```python
print('Enter your name:')
x = input()
print('Hello, ' + x)
```

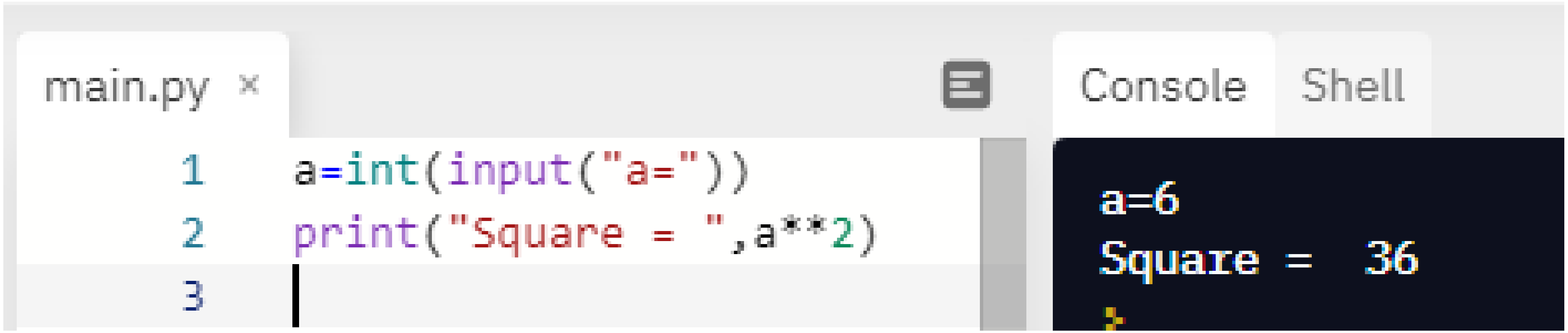
```
Enter your name:
Zhanar
Hello, Zhanar
```

Task for entering data from the keyboard and displaying the result on the screen: **Write a questionnaire program.**

1 task
0,5 point

- To do this, ask the user's first name, last name, age, hobby, favorite subject at school, favorite color, favorite game, favorite movie.
- And output in the format: My name is (surname) + (name), I am (age) years old, I like to do (hobby), my favorite subject is (subject), my favorite color is (color), my favorite game(game), and my favorite movie (movie). It was nice to meet you!

When you work with numbers, you need to convert a string to a number

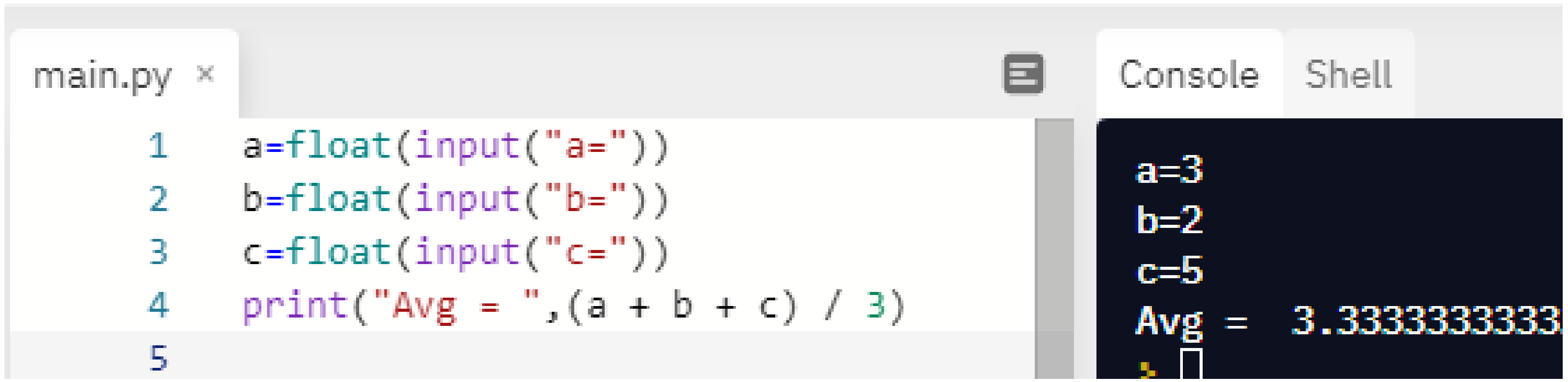


```
a=int(input("a="))
print("Square = ",a**2)
```

```
a=6
Square =  36
>
```

Convert this piece of C# code to Python

```
private void button1_Click(object sender, EventArgs e)
{float a, b, c;
a = Convert.ToSingle(textBox1.Text);
b = Convert.ToSingle(textBox2.Text);
c = Convert.ToSingle(textBox3.Text);
label1.Text = "Среднее значение = "+Convert.ToString((a + b + c) / 3);}
```

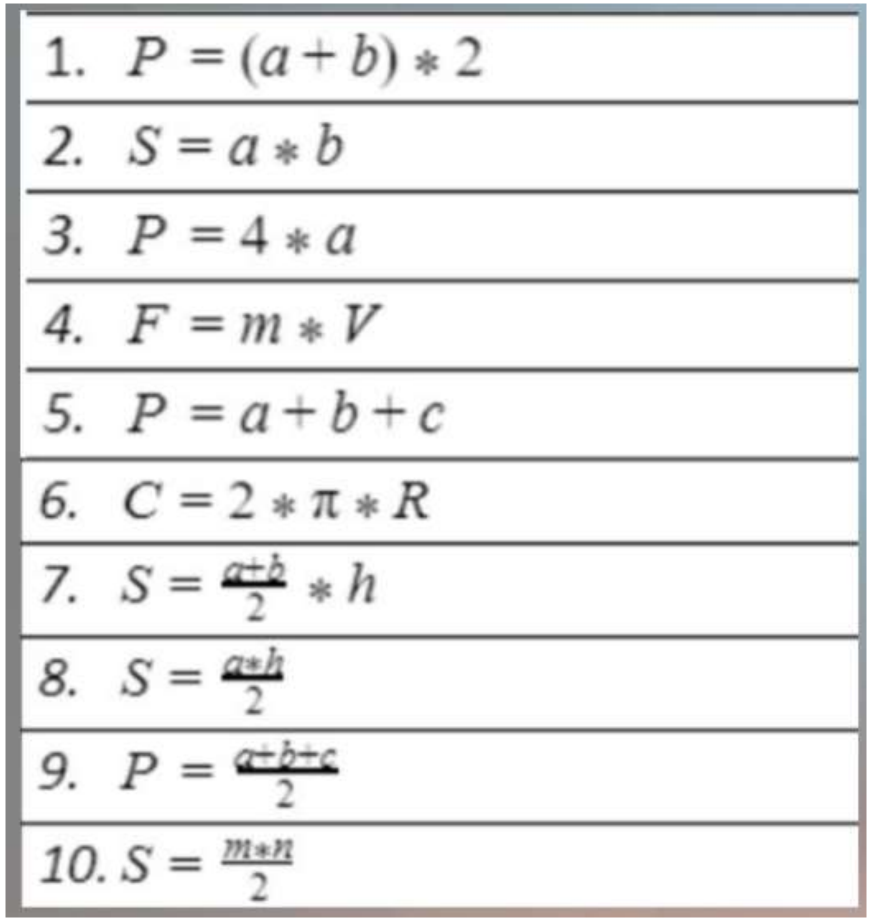


```
a=float(input("a="))
b=float(input("b="))
c=float(input("c="))
print("Avg = ",(a + b + c) / 3)
```

```
a=3
b=2
c=5
Avg =  3.3333333333
```

Write a program that calculates

2 task
0,5 point



1. $P = (a + b) * 2$
2. $S = a * b$
3. $P = 4 * a$
4. $F = m * V$
5. $P = a + b + c$
6. $C = 2 * \pi * R$
7. $S = \frac{a+b}{2} * h$
8. $S = \frac{a*h}{2}$
9. $P = \frac{a+b+c}{2}$
10. $S = \frac{m*n}{2}$