Lesson 3:

Input

Make predictions (think, pair, share)

```
age = 22
print(f"My age is {age}")
```

Questions

What will be the output of `print`, when this program is executed?

1 `My age is age`

[✓ 2] `My age is 22`

3 It is not possible to know the output without executing the program

4 There is an error in the program

True or False? True

This program will always produce the same output, whenever it is executed.

Make predictions (think, pair, share)

```
print(f"My age is {age}")
age = 22
```

Questions

What will be the output of `print`, when this program is executed?

1. `My age is age`
2. `My age is 22`
3. It is not possible to know the output without executing the program

[✓ 4] There is an error in the program

During program execution, a variable must have been assigned a value before that value is referenced.

Make predictions (think, pair, share)

```
age = 22
age = 23
print(f"My age is {age}")
```

Questions

What will be the output of `print`, when this program is executed?

1 `My age is 45`

2 `My age is 22,23`

[✓ 3] `My age is 23`

4 There is an error in the program

A variable can only hold one value at a time. The last value held in age before the `print` statement was 23.

Lesson 4: Input

In this lesson, you will:

- Obtain input from the keyboard in a program
- Differentiate between the data types: integer, real, Boolean, character, string
- Cast variables by calling a function that will return a new value of the desired data type
- Define runtime errors in programs
- Define validation checks

Make a prediction (think, write, pair, share)

```
1  print("What is your name?")
2  name = input()
3  print(f"Hello {name}")
```

Question

Take a look at these three lines of code.

- What do you think each line of code will do when executed?
- What would be the output of line 3?

Walking through the code

```python
print("What is your name?")
name = input()
print(f"Hello {name}")
```

State

Input/Output

Walking through the code

```python
print("What is your name?")
name = input()
print(f"Hello {name}")
```

State

Input/Output

```
What is your name?
```

Walking through the code

```python
print("What is your name?")
name = input()
print(f"Hello {name}")
```

The name **variable** is **initialised** and **assigned** with the user **input**.

State

name

Input/Output

```
What is your name?
```

Walking through the code

```
print("What is your name?")
name = input()
print(f"Hello {name}")
```

State

name

Input/Output

```
What is your name?
Rebecca
```

The name **variable** is **initialised** and **assigned** with the user **input**.

Walking through the code

```python
print("What is your name?")
name = input()
print(f"Hello {name}")
```

State

name "Rebecca"

Input/Output

```
What is your name?
Rebecca
```

The name **variable** is **initialised** and **assigned** with the user **input**.

Walking through the code

```python
print("What is your name?")
name = input()
print(f"Hello {name}")
```

State

name "Rebecca"

Input/Output

```
What is your name?
Rebecca
Hello Rebecca
```

Hello is displayed on the screen with the text that was **input** and **held** in the **name variable**.

Question

How could you use **inputs** to improve your silly story programs?

Adding inputs to your silly stories

Live coding demonstration

ncce.io/sillystory

Activity 1

Improve your own silly stories

Open your silly stories from last lesson and add the **`input`** and **`print`** statements required to make your program more interactive.

Code reminder

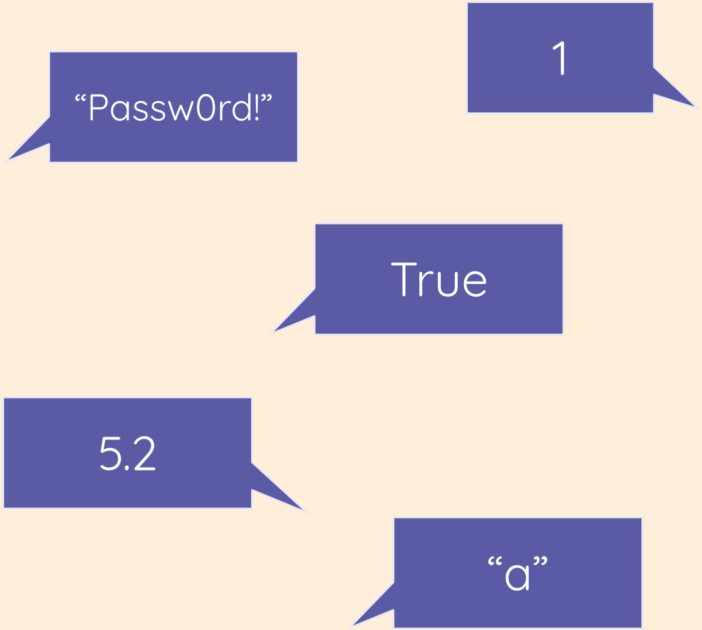
```
print("Enter a colour :")

colour = input()
```

Types of data

Most programming languages will require a **variable** to be **declared** before it is **used**.

Declaring a **variable** means to state what **type** of data will be held by that variable.

Python does not require this and works a little differently.



1

“Passw0rd!”

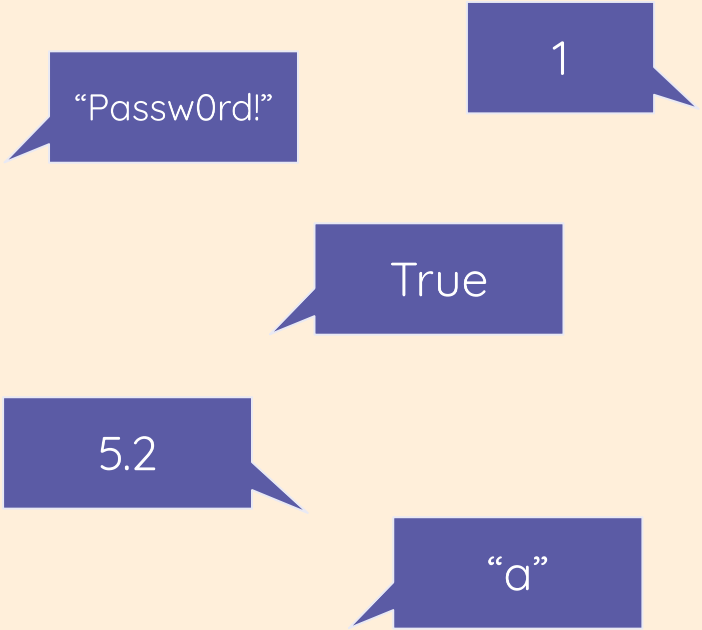
True

5.2

“a”

Types of data

Python doesn’t require you to **declare** a variable. However, you still need to be aware of **data types** because incorrect data types can cause errors in your programs.



1

“Passw0rd!”

True

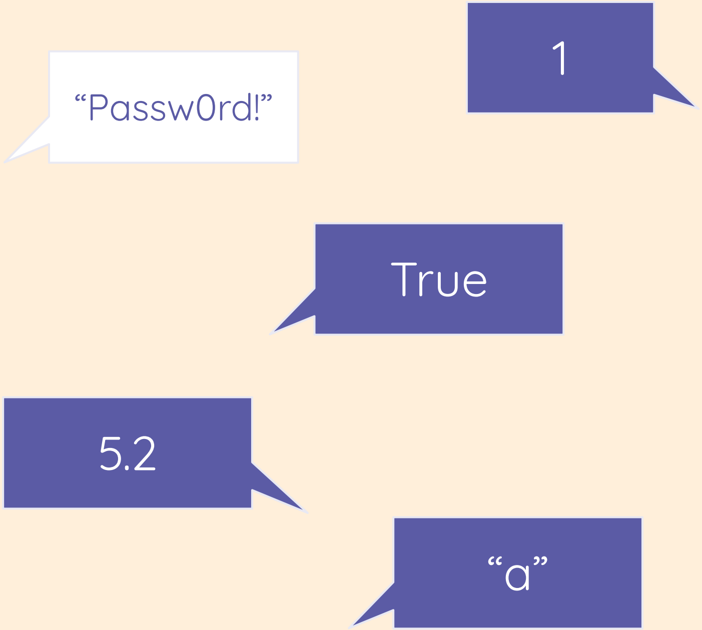
5.2

“a”

Types of data

There are five main data types that you need to be aware of:

- String (text)



1

“Passw0rd!”

True

5.2

“a”

Types of data

There are five main data types that you need to be aware of:

- String (text)
- Integer (whole numbers)

1

“Passw0rd!”

True

5.2

“a”

Types of data

There are five main data types that you need to be aware of:

- String (text)
- Integer (whole numbers)
- Boolean (True or False)

1

“Passw0rd!”

True

5.2

“a”

Types of data

There are five main data types that you need to be aware of:

- String (text)
- Integer (whole numbers)
- Boolean (True or False)
- Real or floating point numbers (decimal numbers)

1

“Passw0rd!”

True

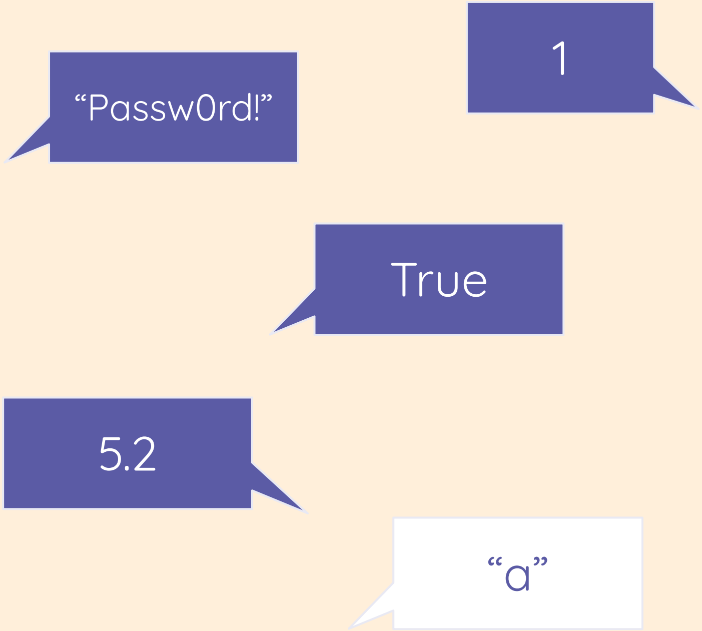
5.2

“a”

Types of data

There are five main data types that you need to be aware of:

- String (text)
- Integer (whole numbers)
- Boolean (True or False)
- Real or floating point numbers (decimal numbers)
- Char (single string characters)



1

“Passw0rd!”

True

5.2

“a”

Types of data

Incorrect data types can cause problems during the execution of your programs.

Question

Predict what might happen when this code is executed.

```
print("Enter a number")
num1 = input()
print("Enter another number")
num2 = input()
print(num1+num2)
```

Types of data

The data type for an **input** is always **string**. When you add two pieces of string together, it will **concatenate** (join) them.

Instead of **adding** the two numbers together to make **3,** it has **joined** the corresponding strings together to make **12** (one,two).

This code has produced a **logic error** because it hasn’t executed as expected.

```
print("Enter a number")
num1 = input()
print("Enter another number")
num2 = input()
print(num1+num2)
```

```
Enter a number
1
Enter another number
2
12
>>>
```

Types of data

If you want Python to use your value as an **integer,** then you need to tell it that by **casting** the value.

You do this by placing **`input()`** inside the **`int()`** function.

```
print("Enter a number")
num1 = int(input())
print("Enter another number")
num2 = int(input())
print(num1+num2)
```

```
Enter a number
1
Enter another number
2
3
>>>
```

Types of data

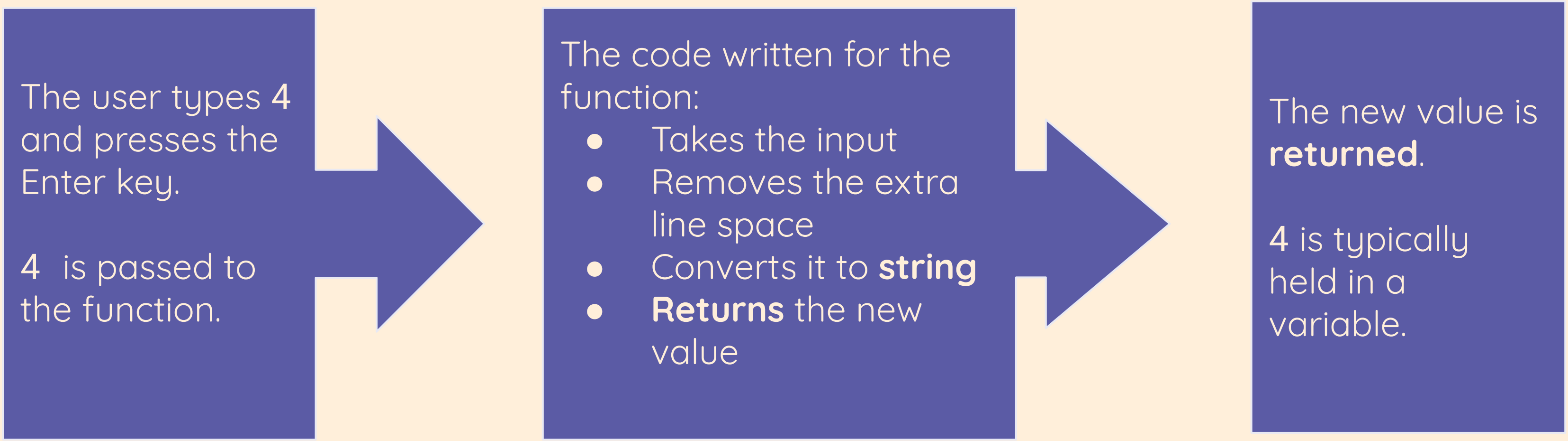
`input()` and **`int()`** are both **functions**.

They are a type of **subroutine** that takes a value, processes it, and then **returns** another value back.

```
print("Enter a number")
num1 = int(input())
print("Enter another number")
num2 = int(input())
print(num1+num2)
```

```
Enter a number
1
Enter another number
2
3
>>>
```

Input, process, and output of `input()`



Input

The user types 4 and presses the Enter key.

4 is passed to the function.

Process

The code written for the function:

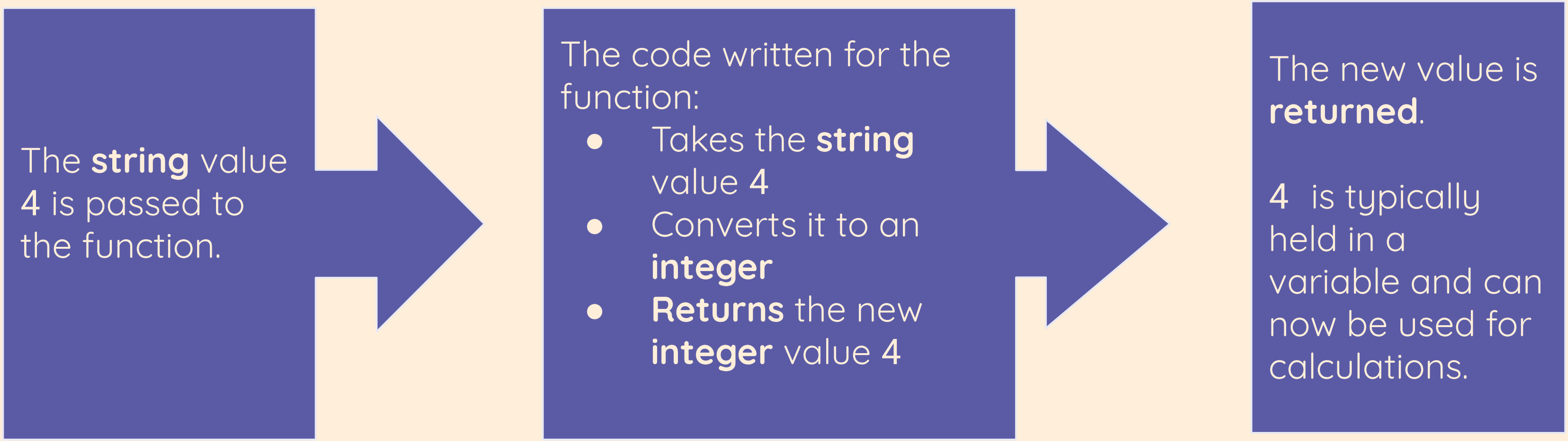
- Takes the input
- Removes the extra line space
- Converts it to **string**
- **Returns** the new value

Output

The new value is **returned**.

4 is typically held in a variable.

Input, process, and output of `int()`



Input

The **string** value 4 is passed to the function.

Process

The code written for the function:

- Takes the **string** value 4
- Converts it to an **integer**
- **Returns** the new **integer** value 4

Output

The new value is **returned**.

4 is typically held in a variable and can now be used for calculations.

Types of data

Converting a value from one data type to another is known as **casting**.

```
print("Enter a number")
num1 = int(input())
print("Enter another number")
num2 = int(input())
print(num1+num2)
```

Types of data

Errors can still happen during execution, even when casting has been used.

Question

What might happen if the user enters ‘four’ when this code is executed?

```
print("Enter a number")
num1 = int(input())
print("Enter another number")
num2 = int(input())
print(num1+num2)
```

Types of data

Answer

A **runtime error** occurs. This is a type of error that causes the program to crash during execution.

```
print("Enter a number")
number = int(input())
print(number)
```

```
Enter a number
four
Traceback (most recent call last):
  File "c:\users\pi\mu_code\fsea.py", line
2, in <module>
    number = int(input())
ValueError: invalid literal for int() with
base 10: 'four'
>>>
```

Types of data

You can avoid this type of error by introducing **validation checks**.

Here is an example check that you can use called `try` and `except`.

You will learn more about these later on in the course.

```python
print("Enter a number")
try:
    number = int(input())
except ValueError:
    print("You must enter a number")
    number = int(input())
```

Types of data

To convert values to different data types, you need to know the **functions** that are available to you.

Here are the most common functions that you will need to know.

You can find these in the Python documentation.

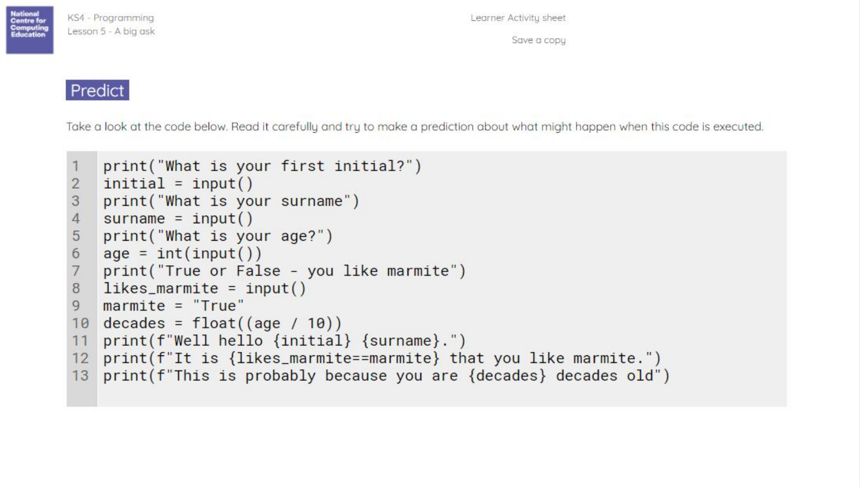
ncce.io/pythonfunctions

```
# convert to string

str()

# convert to integer

int()

# convert to real

float()
```

Mini data collection program

Use the **worksheet** to predict, run, and investigate the code for a mini data collection program.

An **explorer task** has been provided on the sheet if you require this.



KS4 - Programming
Lesson 5 - A big ask

Learner Activity sheet

Save a copy

Predict

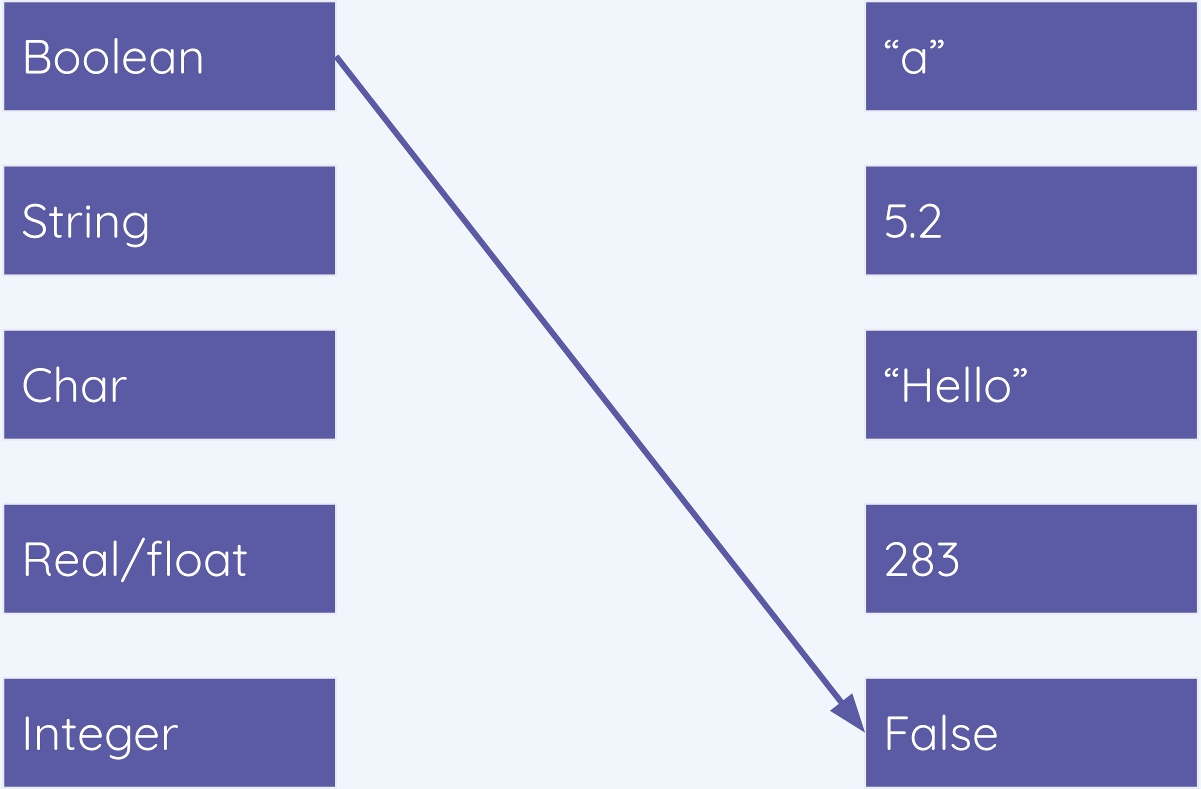
Take a look at the code below. Read it carefully and try to make a prediction about what might happen when this code is executed.

```
print("What is your first initial?")
initial = input()
print("What is your surname")
surname = input()
print("What is your age?")
age = int(input())
print("True or False - you like marmite")
likes_marmite = input()
marmite = "True"
decades = float((age / 10))
print(f"Well hello {initial} {surname}.")
print(f"It is {likes_marmite==marmite} that you like marmite.")
print(f"This is probably because you are {decades} decades old")
```

Match the data types to their examples

Boolean	“a”
String	5.2
Char	“Hello”
Real/float	283
Integer	False

Match the data types to their examples



Boolean

String

Char

Real/float

Integer

“a”

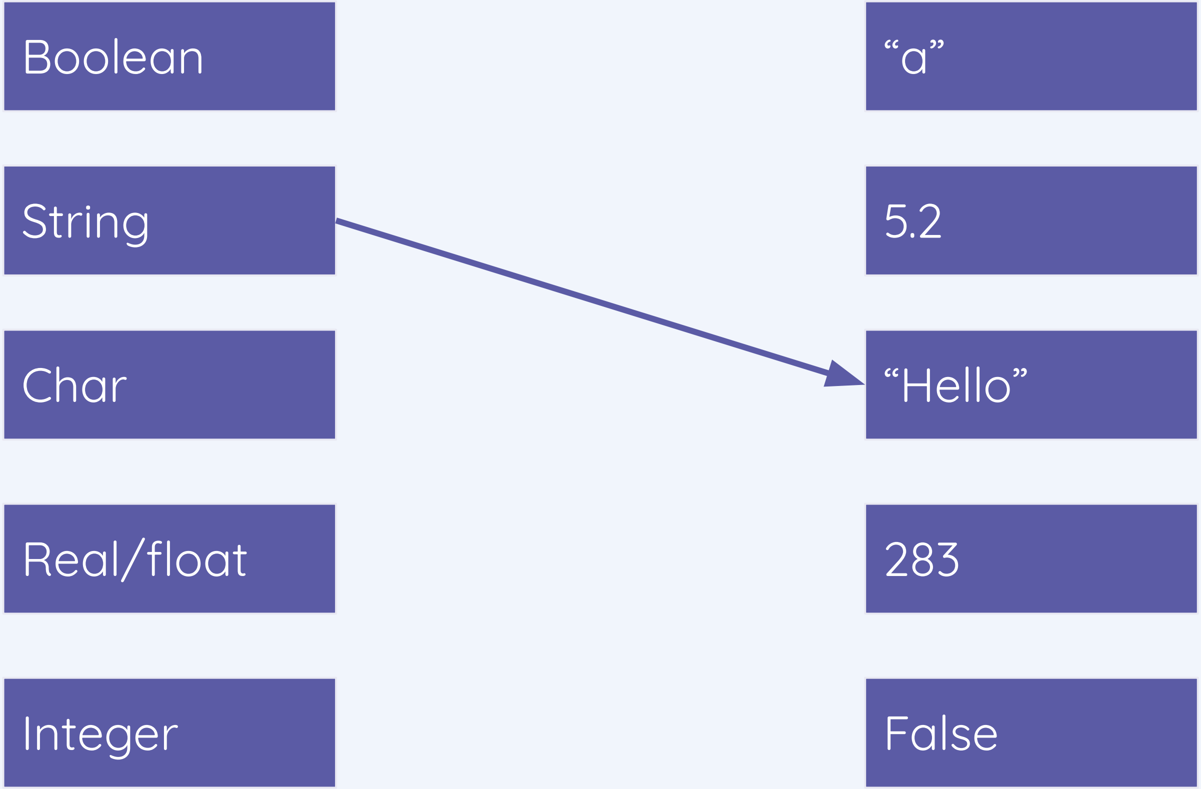
5.2

“Hello”

283

False

Match the data types to their examples



Boolean

String

Char

Real/float

Integer

“a”

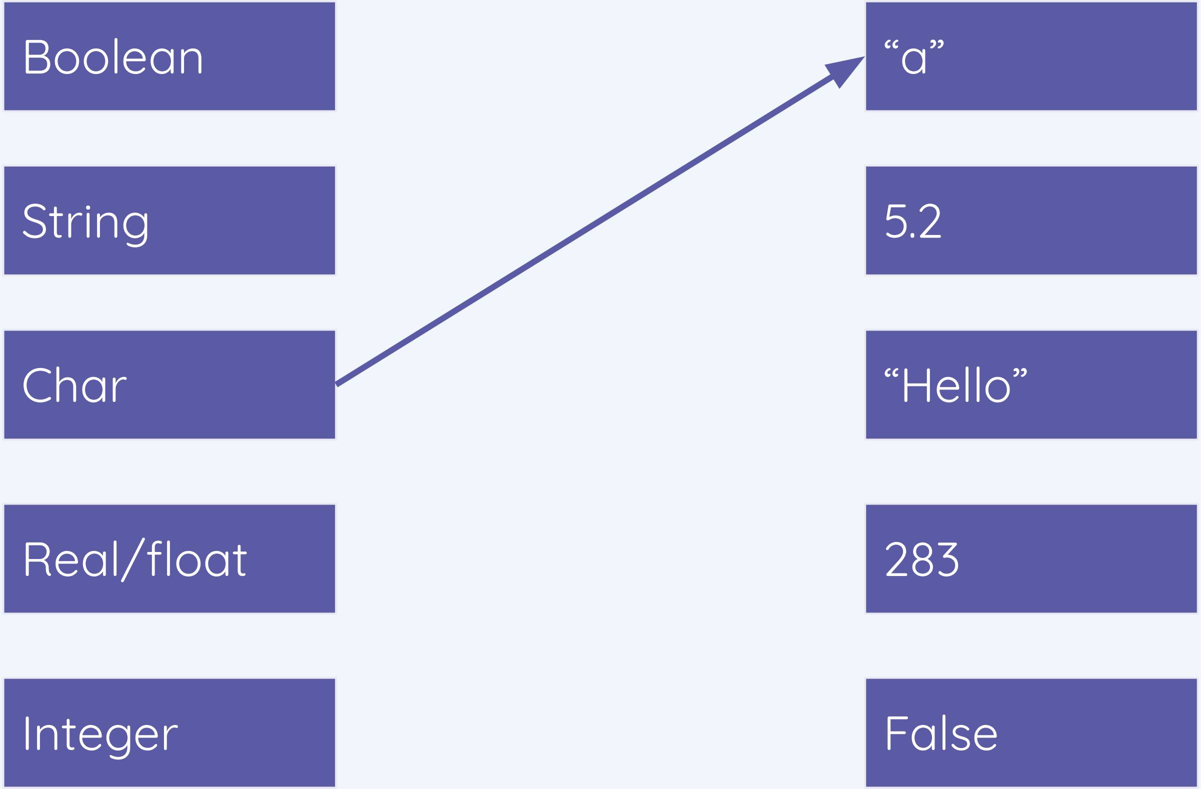
5.2

“Hello”

283

False

Match the data types to their examples



Boolean

String

Char

Real/float

Integer

“a”

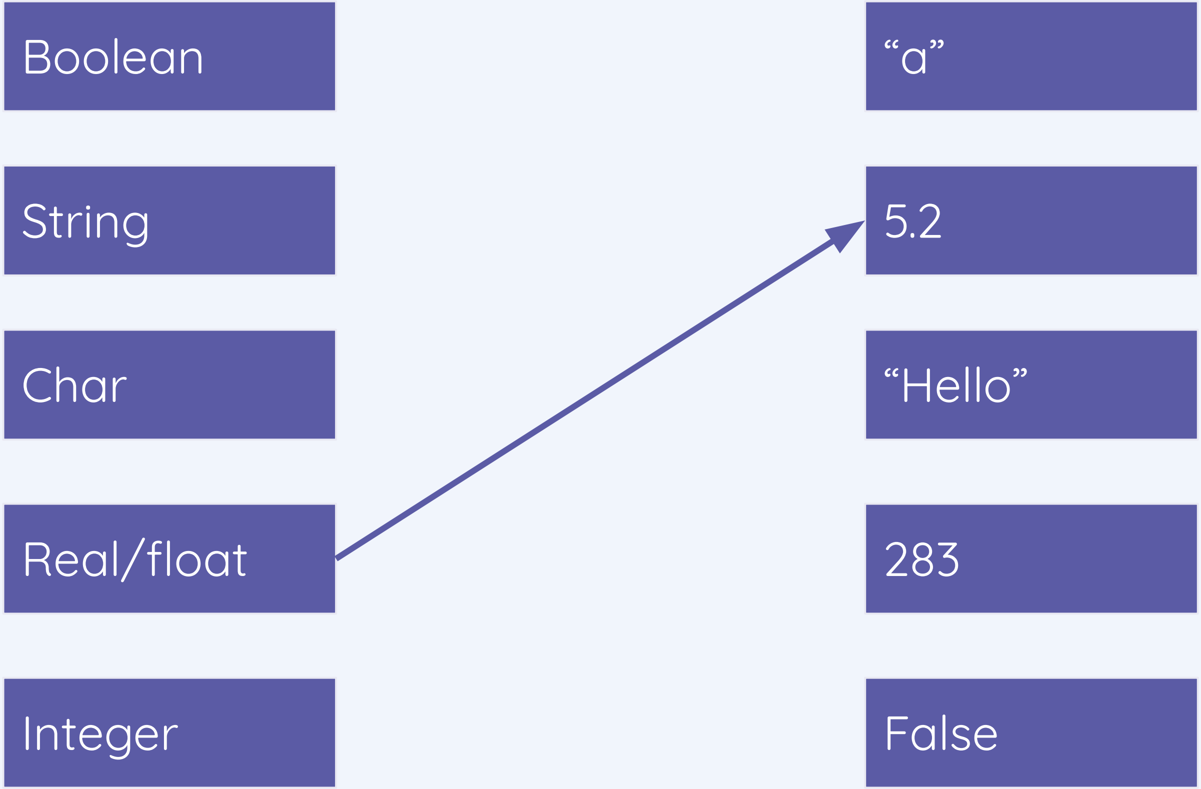
5.2

“Hello”

283

False

Match the data types to their examples



Boolean

String

Char

Real/float

Integer

“a”

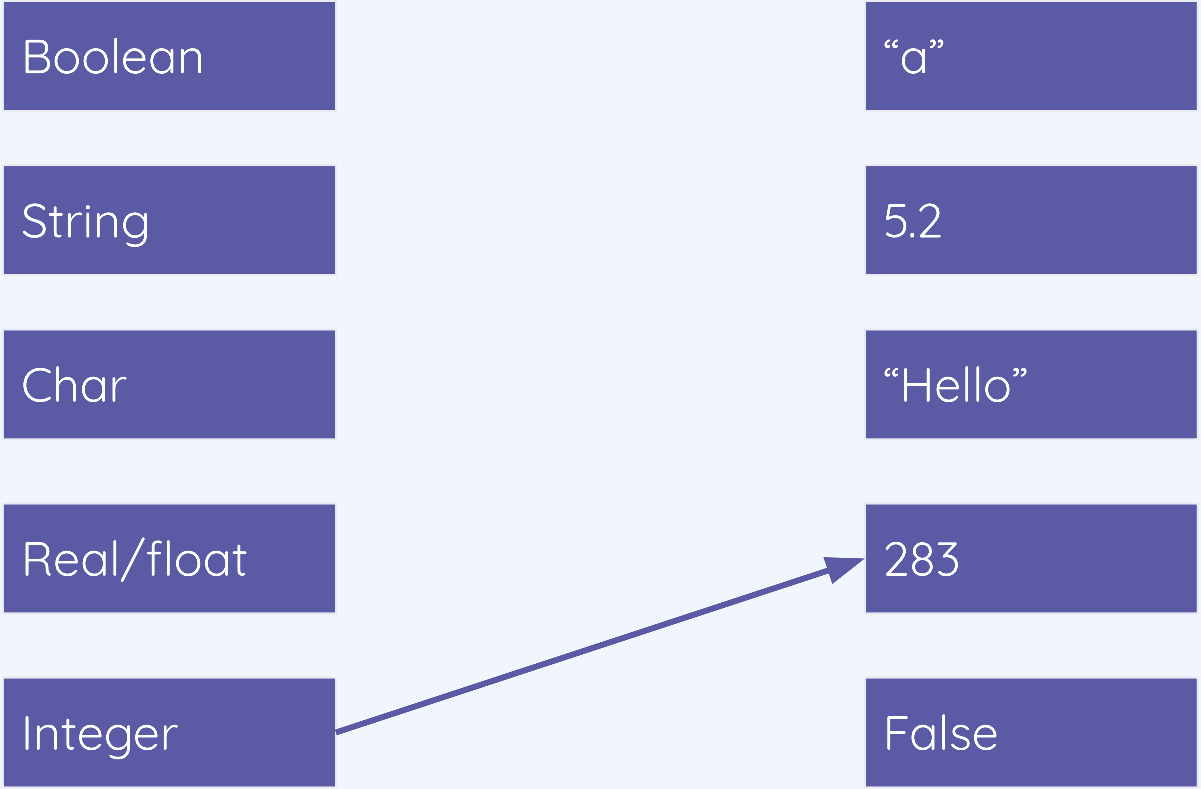
5.2

“Hello”

283

False

Match the data types to their examples



Boolean

String

Char

Real/float

Integer

“a”

5.2

“Hello”

283

False

Next lesson

In this lesson, you…

Learnt how to implement input in your programs

Discovered data types and how these work in Python

Next lesson, you will…

Learn about random number generation