Python 3 Fundamentals

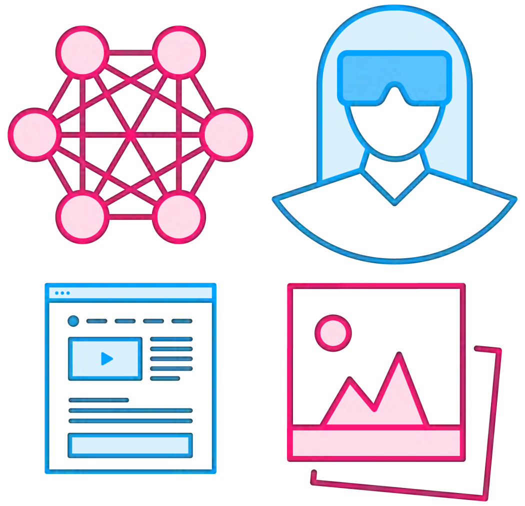
Data Types and Input and Output

Sarah Holderness

Author

@dr_holderness

Why Python?

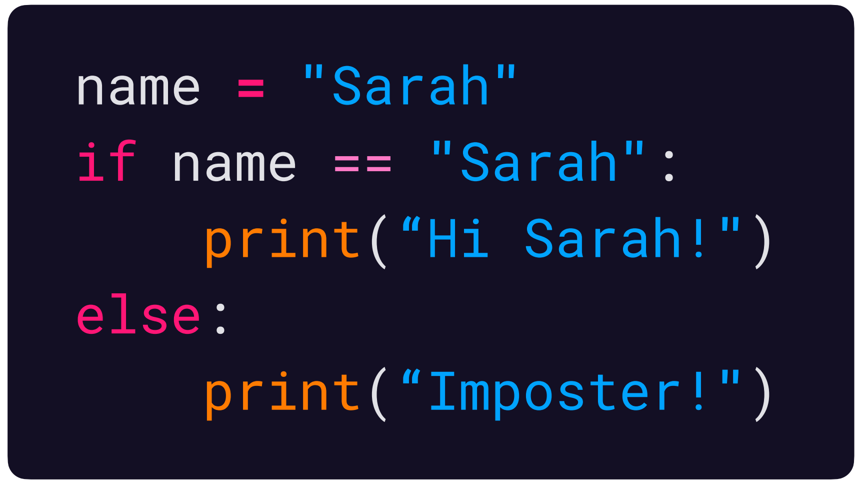


Versatile

Data science, machine learning, web development, & more

Strong Community

There’s a package for everything

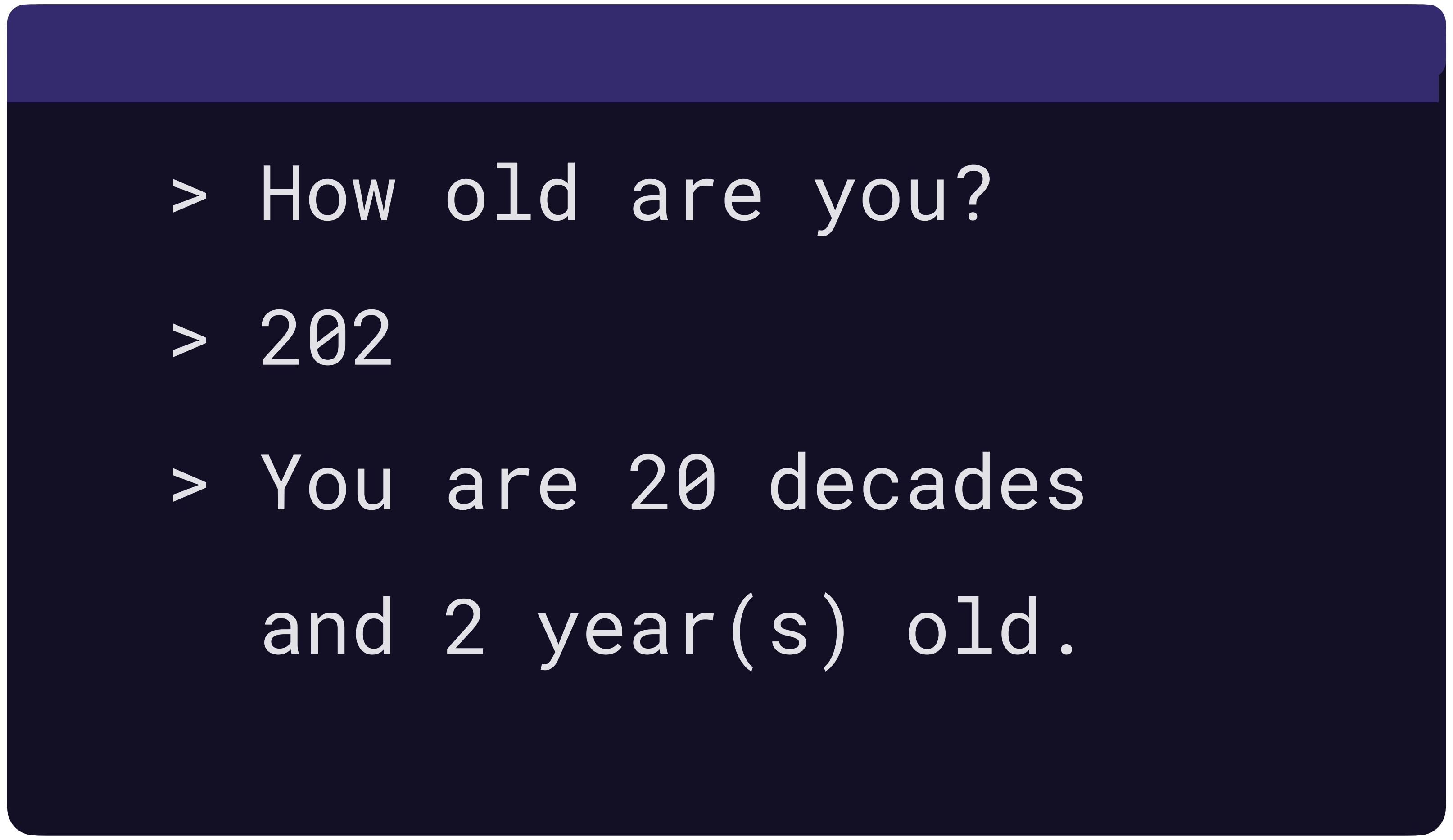


```
name = "Sarah"
if name == "Sarah":
    print(“Hi Sarah!")
else:
    print(“Imposter!")
```

Easy to Learn

Easy-to-read, concise, interpreted language

Where Do We Start?



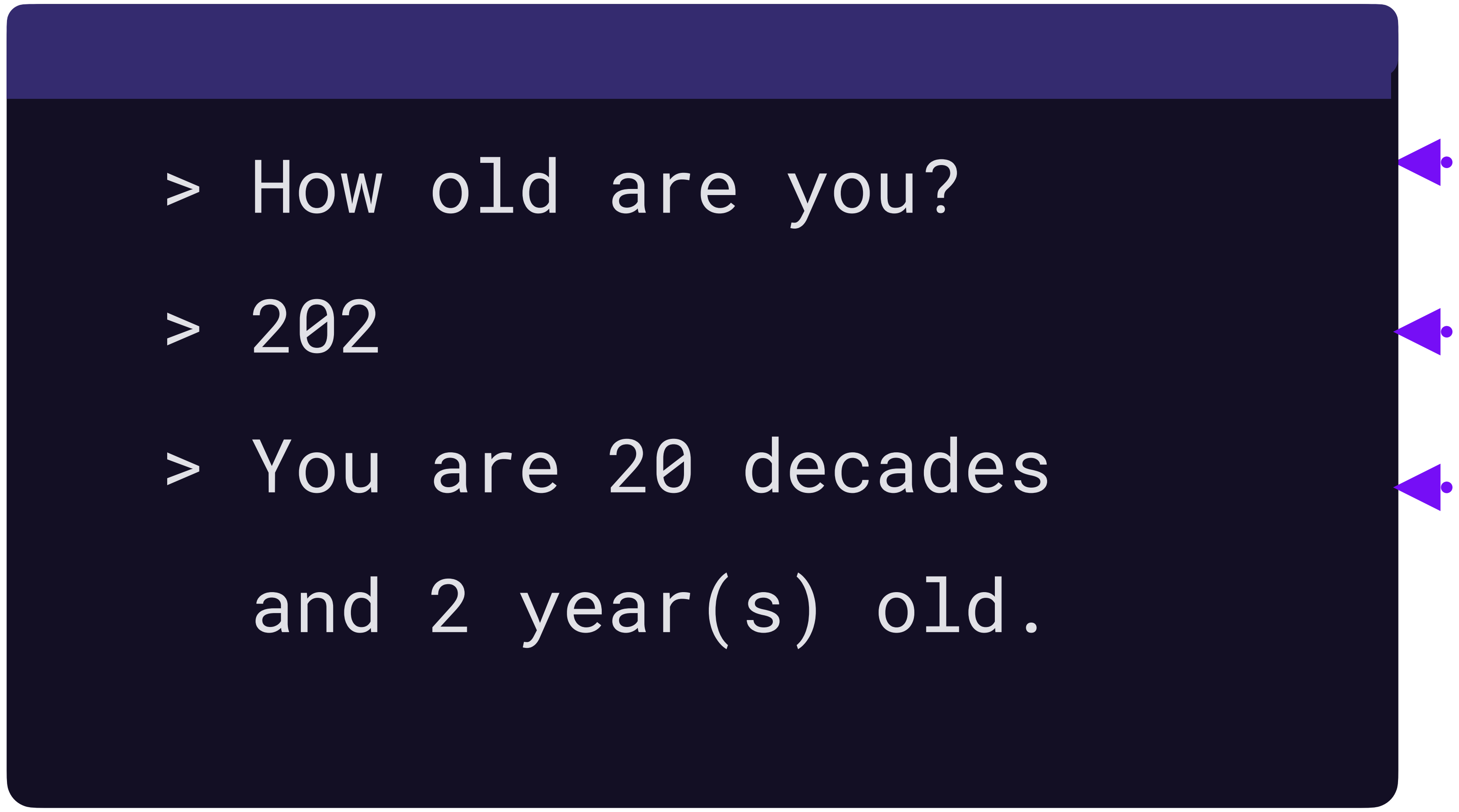
```
> How old are you?

> 202

> You are 20 decades

  and 2 year(s) old.
```

Where Do We Start?



```
> How old are you?

> 202

> You are 20 decades

  and 2 year(s) old.
```

Ask the user for input

Save the input to a variable

Calculate the decades and years

Convert these numbers to text

Print the result to the screen

Where Do We Start?

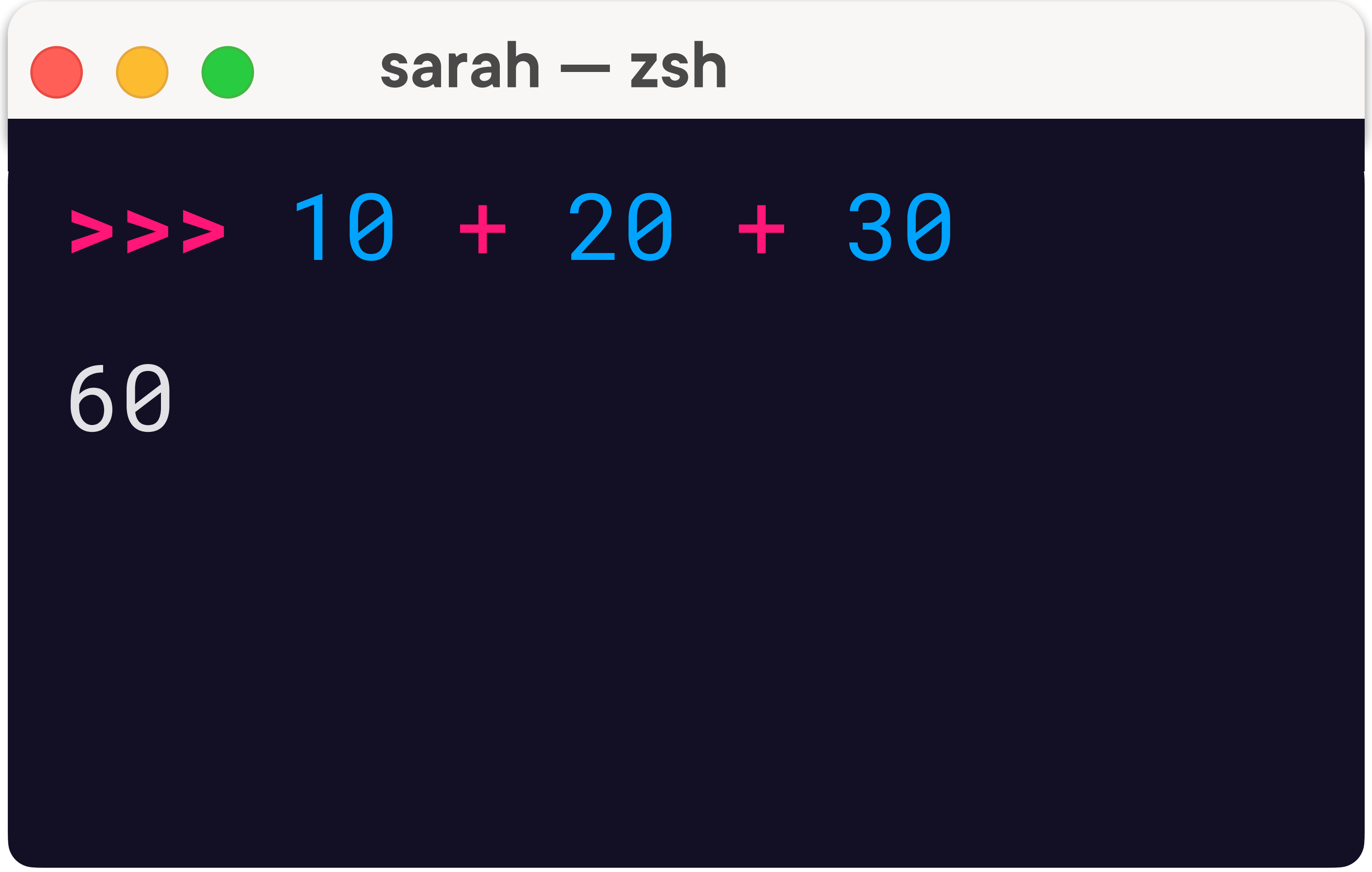
In This Video

- We'll save numbers to variables
- Do basic math calculations, like calculate the area of a room

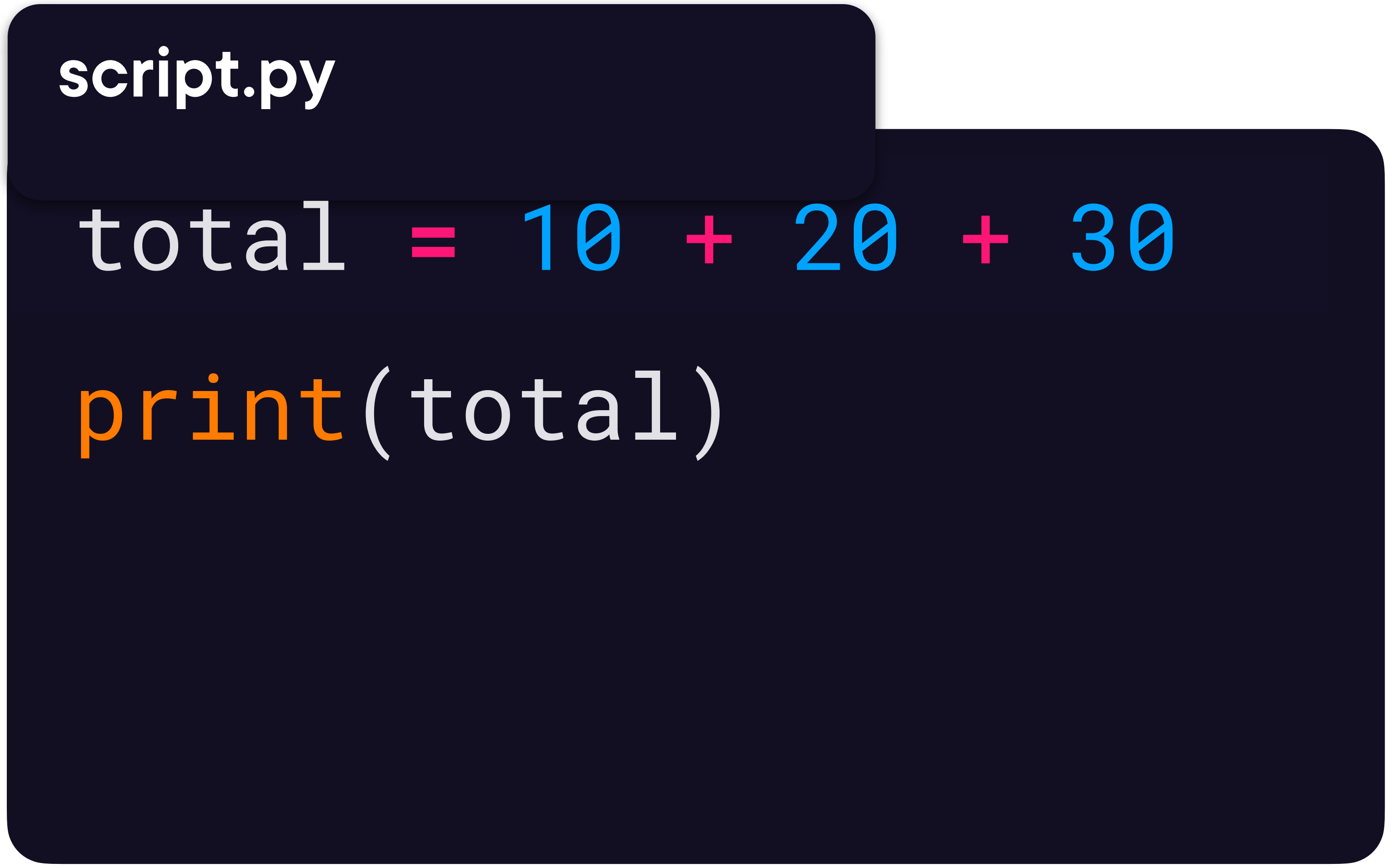
You Can

- Follow along if you want
- (Or wait until the next clip to install Python)

Where Do We Write Python Code?



```
>>> 10 + 20 + 30
60
```



```python
total = 10 + 20 + 30
print(total)
```

The Python Interactive Shell

The Python shell let's you run Python lines of code one at a time

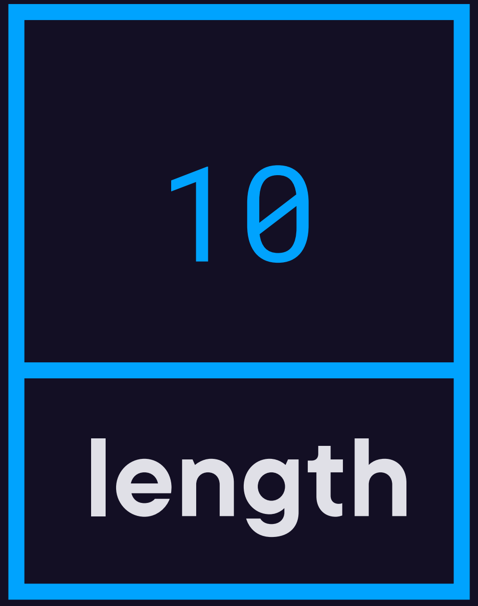
A Python File

A Python script or file is where you create longer Python programs

Saving Numbers to Variables

Assigning the value 10 *to the variable* length

```
>>> length = 10
```

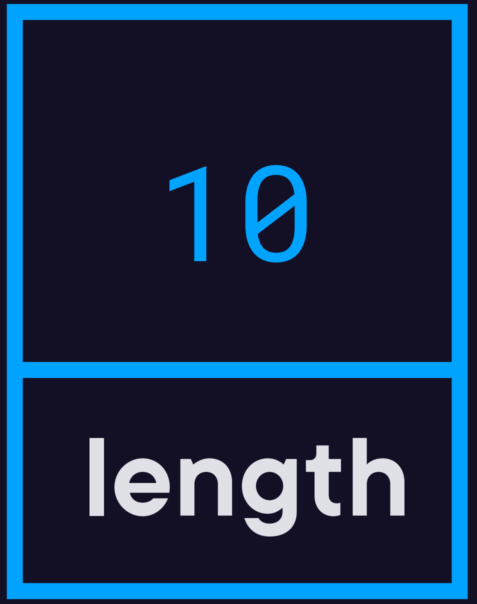


Now on your computer there is a piece of memory labeled length *that stores the value* 10

Saving Numbers to Variables

```
>>> length = 10
>>> length
10
```

From the shell we can enter the name of the variable `length` *to see it’s value and see that it’s actually* `10`



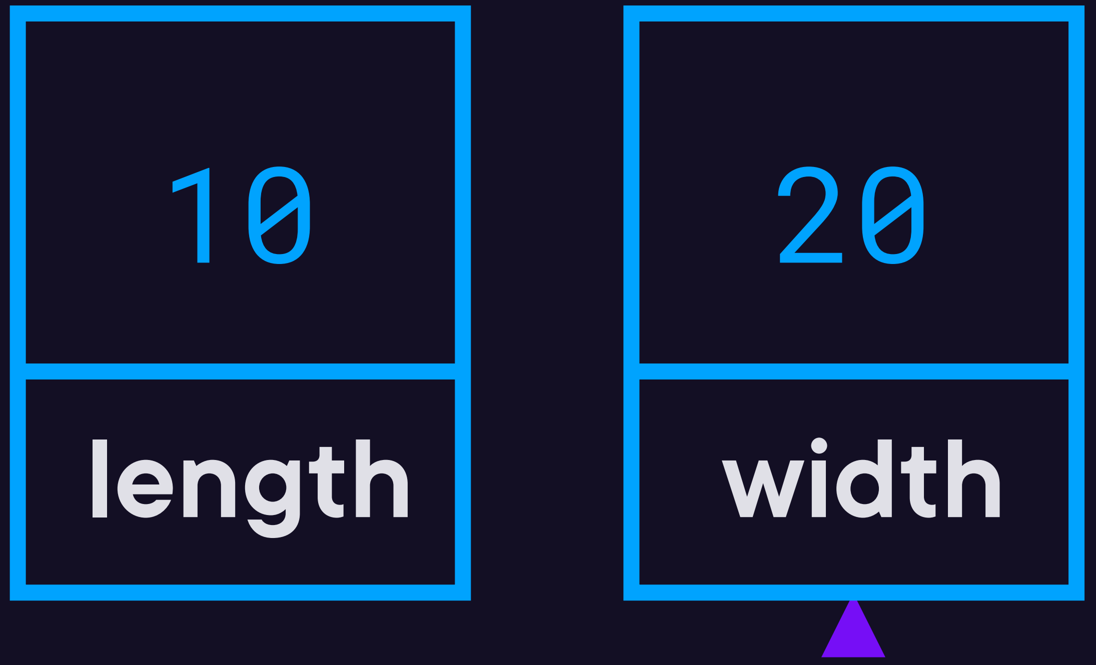
10

length

Saving Numbers to Variables

```
>>> length = 10
>>> width = 20
```

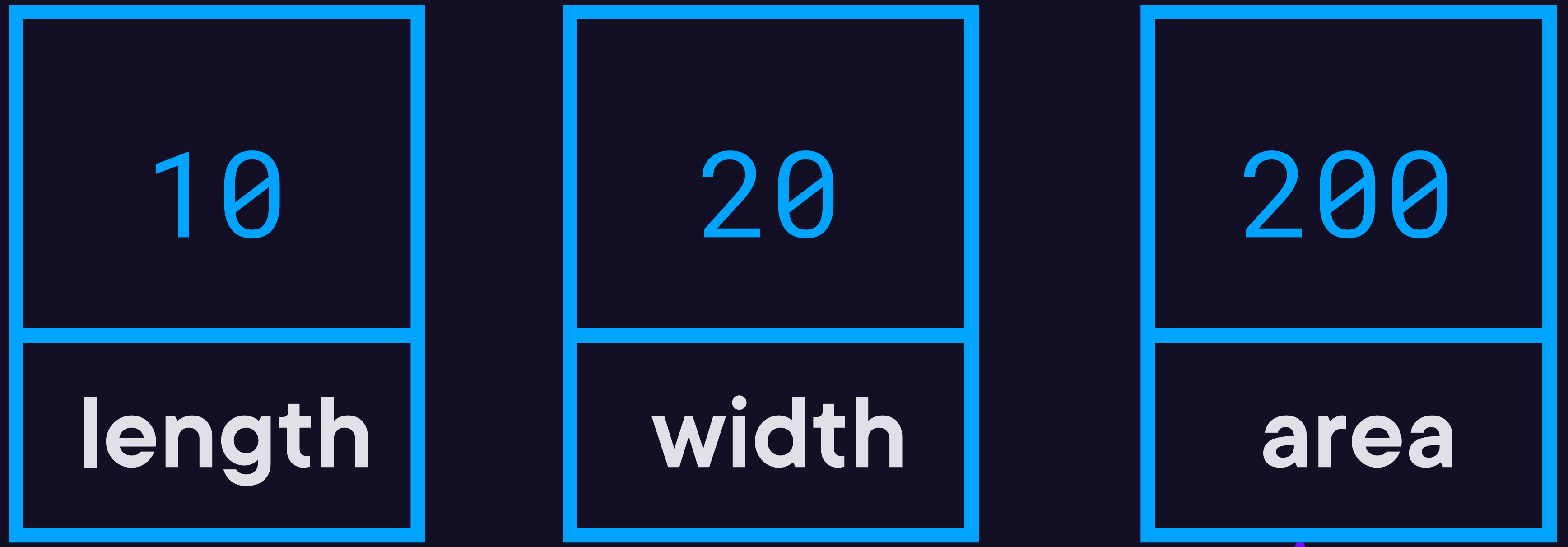
Let’s also add the `width` of the rectangle



Now we have another variable stored in memory

Saving Numbers to Variables

```
>>> length = 10
>>> width = 20
>>> area = length * width
```



10 length

20 width

200 area

Now we can calculate the area with the multiplication operator

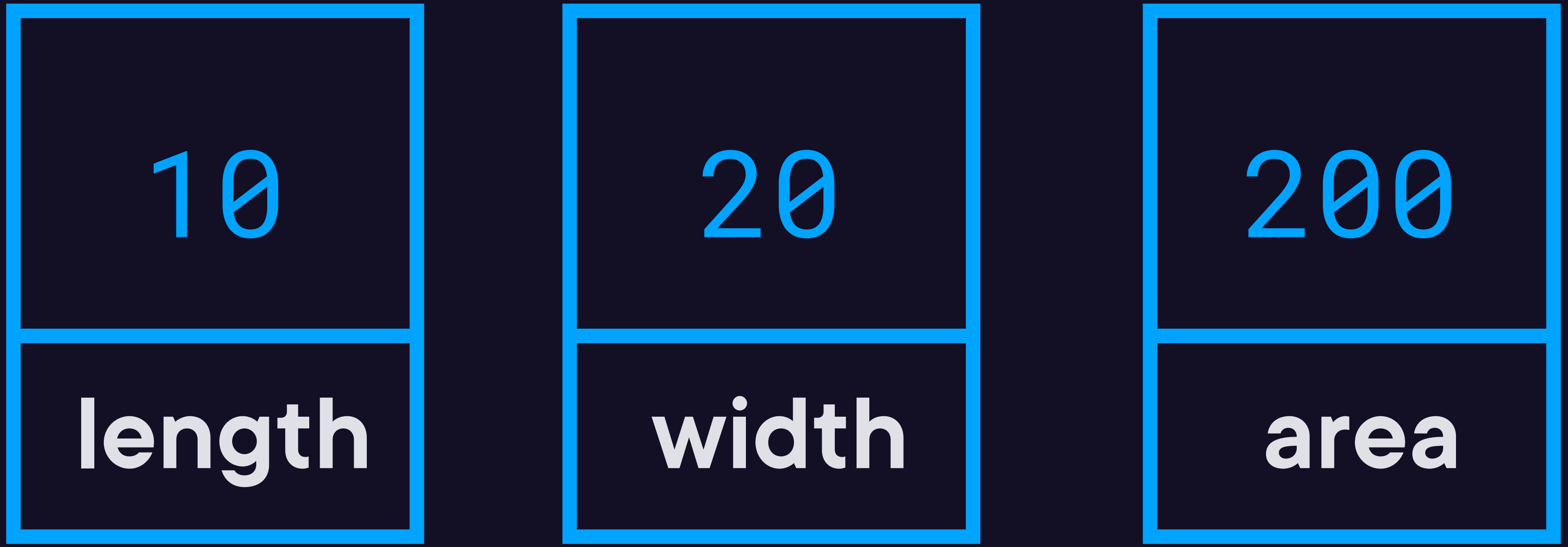
And now we have another variable stored in memory

The arithmetic operators in Python are mostly the same ones you know already from a calculator: + – * /

Saving Numbers to Variables

```
>>> length = 10
>>> width = 20
>>> area = length * width
>>> area
200
```

The value of area is output to the screen



10

length

20

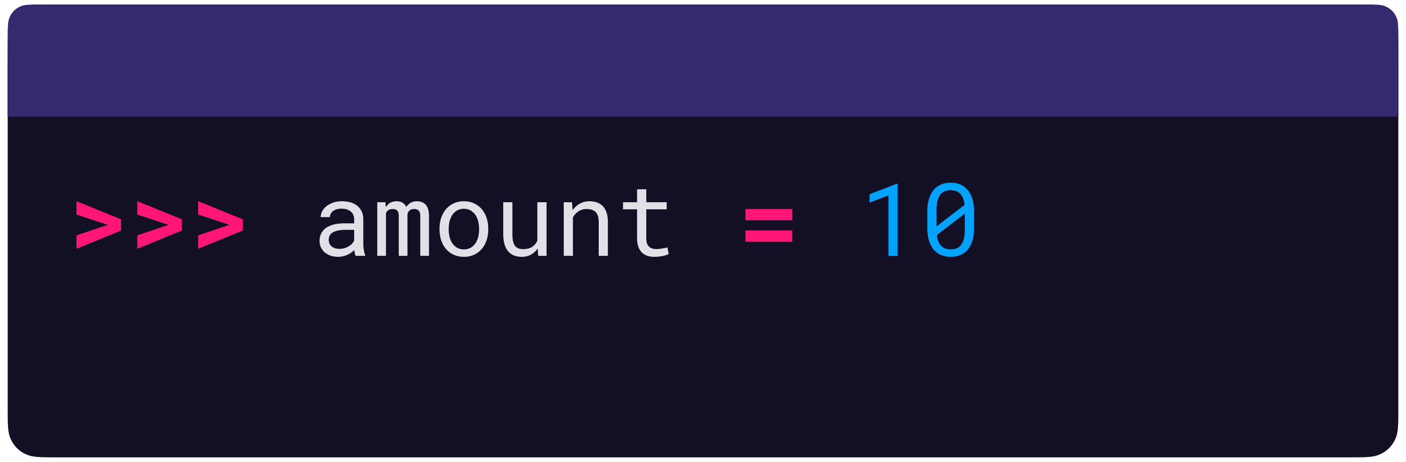
width

200

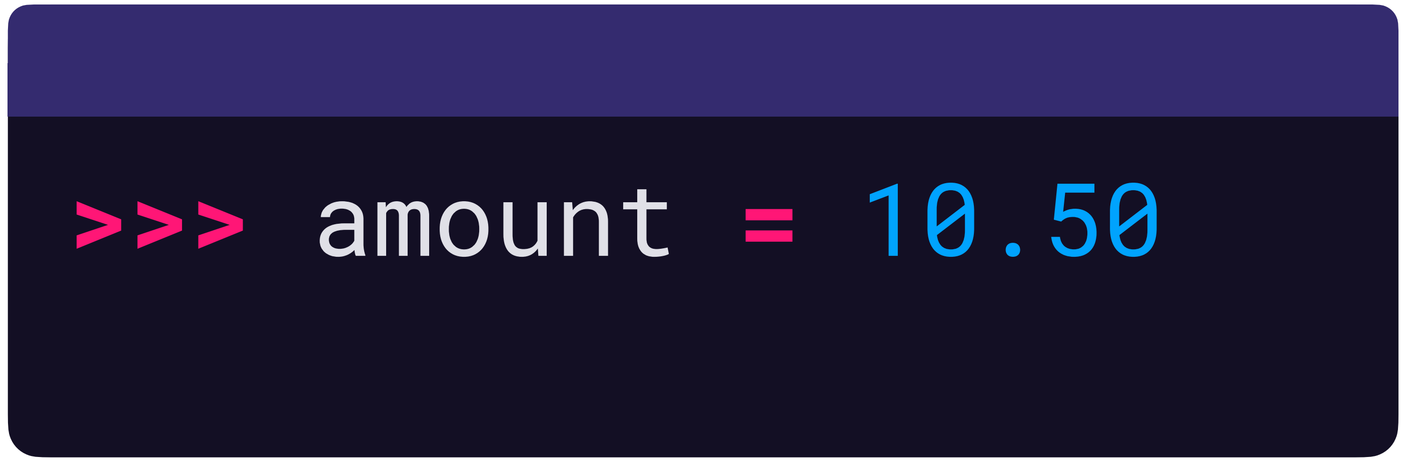
area

Primitive Data Types

Python assumes the type of variable based on the assigned value



```
>>> amount = 10
```



```
>>> amount = 10.50
```

int

Python infers that `amount` is an `int` since it is a whole number

float

Python infers that `amount` is a `float` since it is a decimal

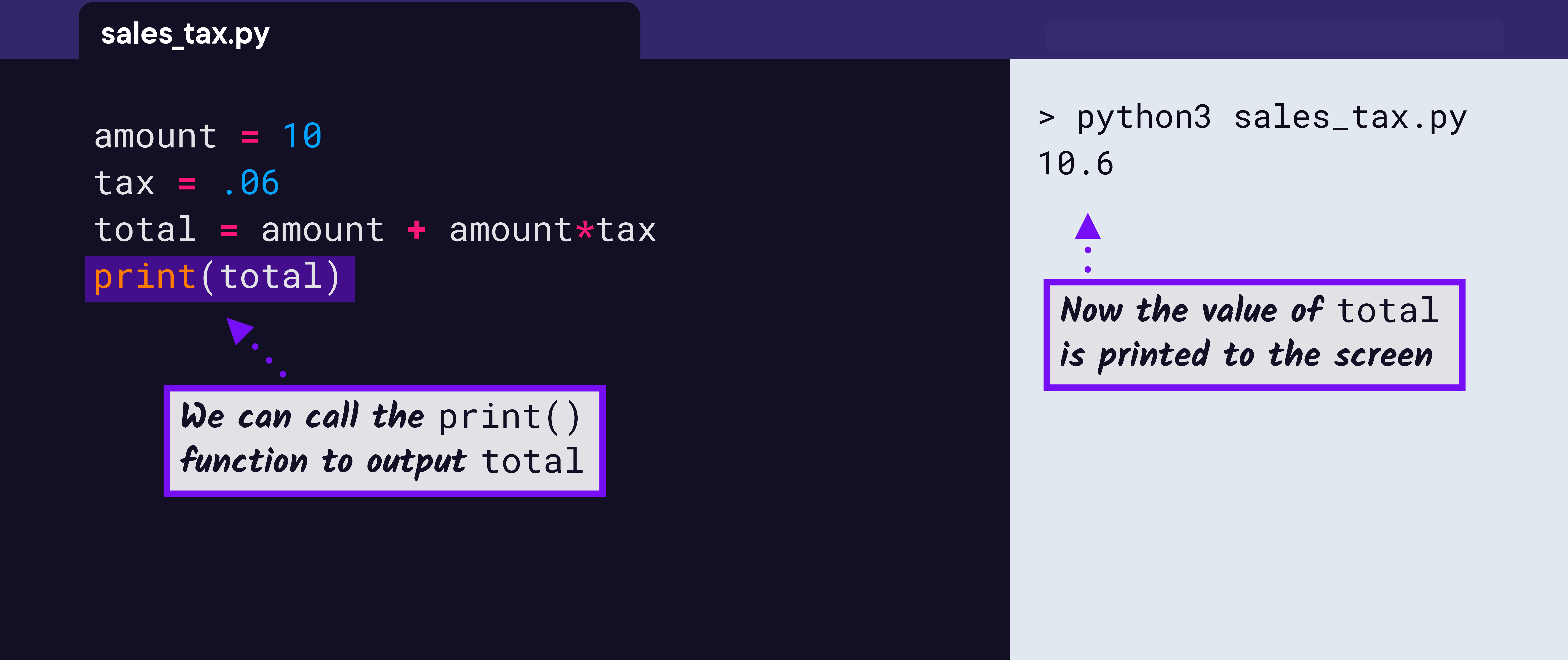
Strings and Input and Output

Sarah Holderness

Author

@dr_holderness

A Python Script



sales_tax.py

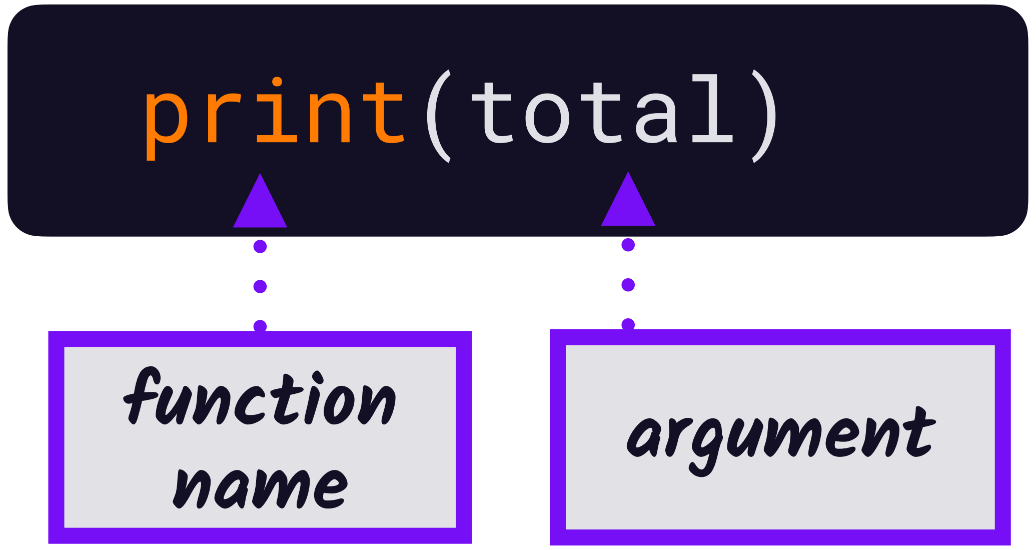
```python
amount = 10
tax = .06
total = amount + amount*tax
print(total)
```

We can call the `print()` *function to output* `total`

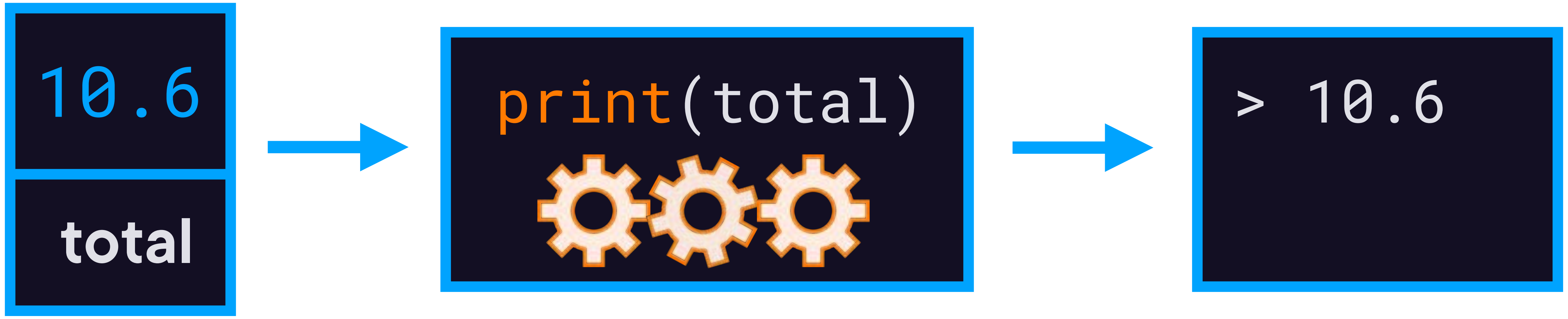
```
> python3 sales_tax.py
10.6
```

Now the value of `total` *is printed to the screen*

Python print() Function



```
print(total)
```

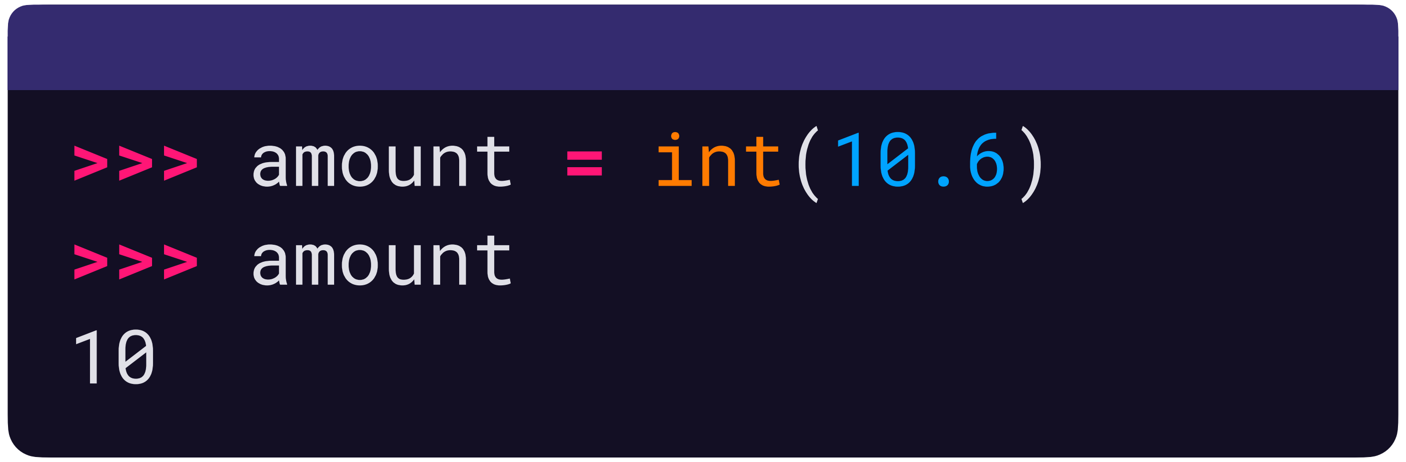


You can think of this function as a black box machine. We don't know how it works...

But we do know it will output the given value to the screen.

Data Type Conversion Functions

What if we want to convert a float to an int?

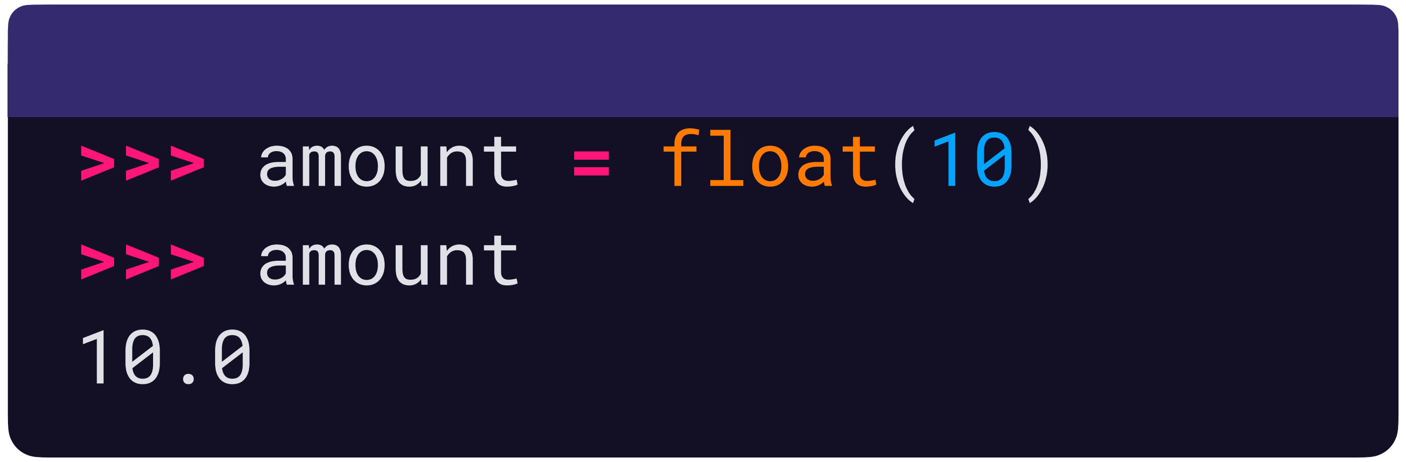


```
>>> amount = int(10.6)
>>> amount
10
```

int()

Use the int() conversion function

What if we want to convert an int to a float?

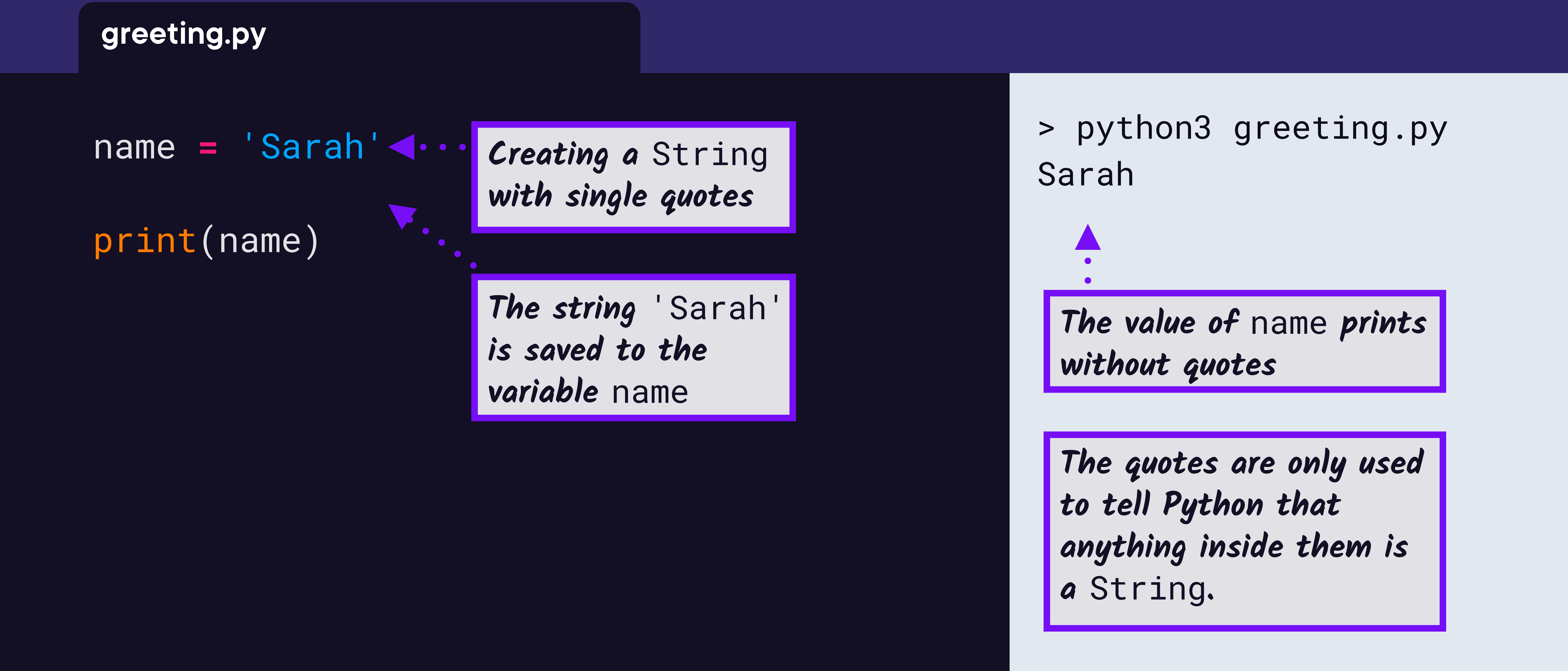


```
>>> amount = float(10)
>>> amount
10.0
```

float()

Use the float() conversion function

A String Stores Text



greeting.py

```python
name = 'Sarah'

print(name)
```

Creating a String with single quotes

The string 'Sarah' is saved to the variable name

```
> python3 greeting.py
Sarah
```

The value of name prints without quotes

The quotes are only used to tell Python that anything inside them is a String.

Create Strings with Single or Double Quotes

greeting.py

```
store_name = "Sarah's Store"
print(store_name)
```

Double quotes are useful if a single quote is literally part of the String

```
store_name = 'Sarah's Store'
print(store_name)
```

This would cause an error because the second single quote would end the String *and Python doesn't know what to do with the rest.*

String Concatenation

greeting.py

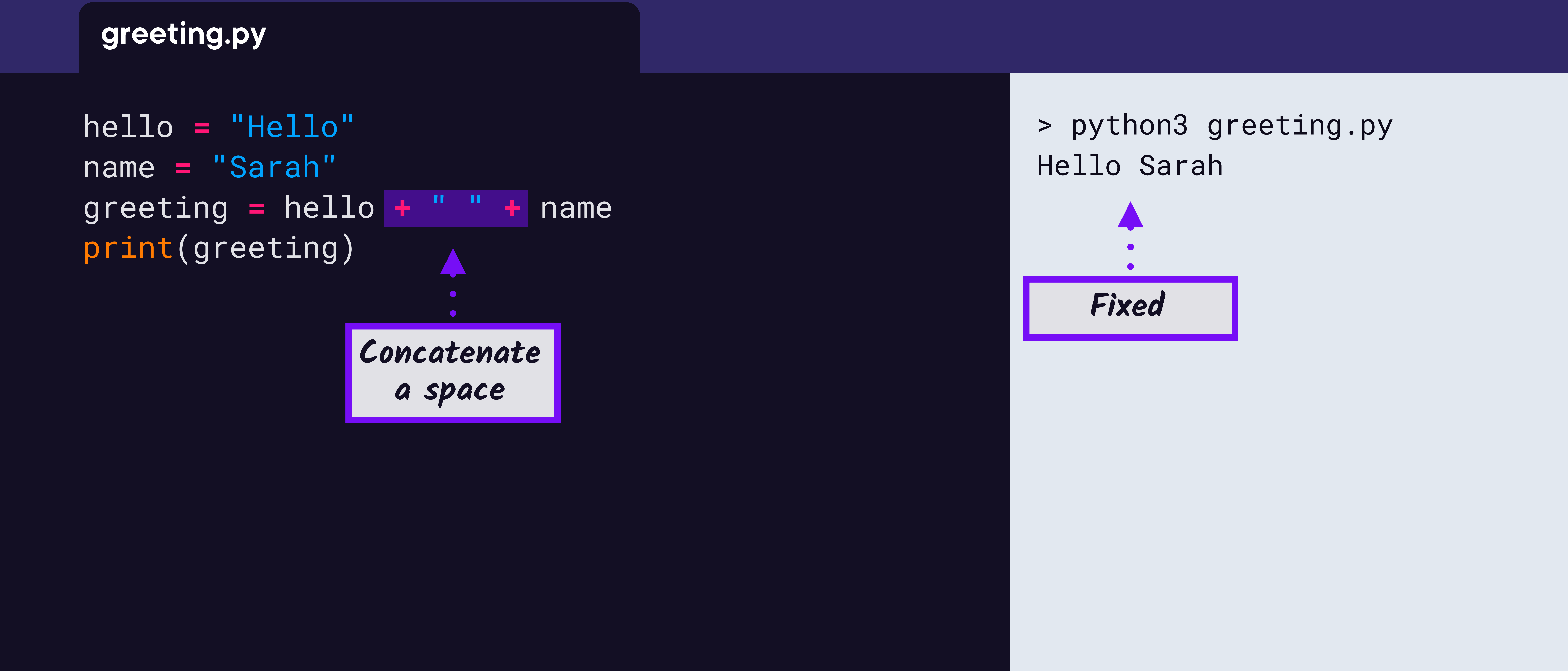
```python
hello = "Hello"
name = "Sarah"
greeting = hello + name
print(greeting)
```

Concatenate two Strings with a +

```
> python3 greeting.py
HelloSarah
```

Notice how the two strings are smushed together? We need a space between them.

Fixing Our Program



greeting.py

```python
hello = "Hello"
name = "Sarah"
greeting = hello + " " + name
print(greeting)
```

Concatenate a space

```
> python3 greeting.py
Hello Sarah
```

Fixed

Fixing Our Program

greeting.py

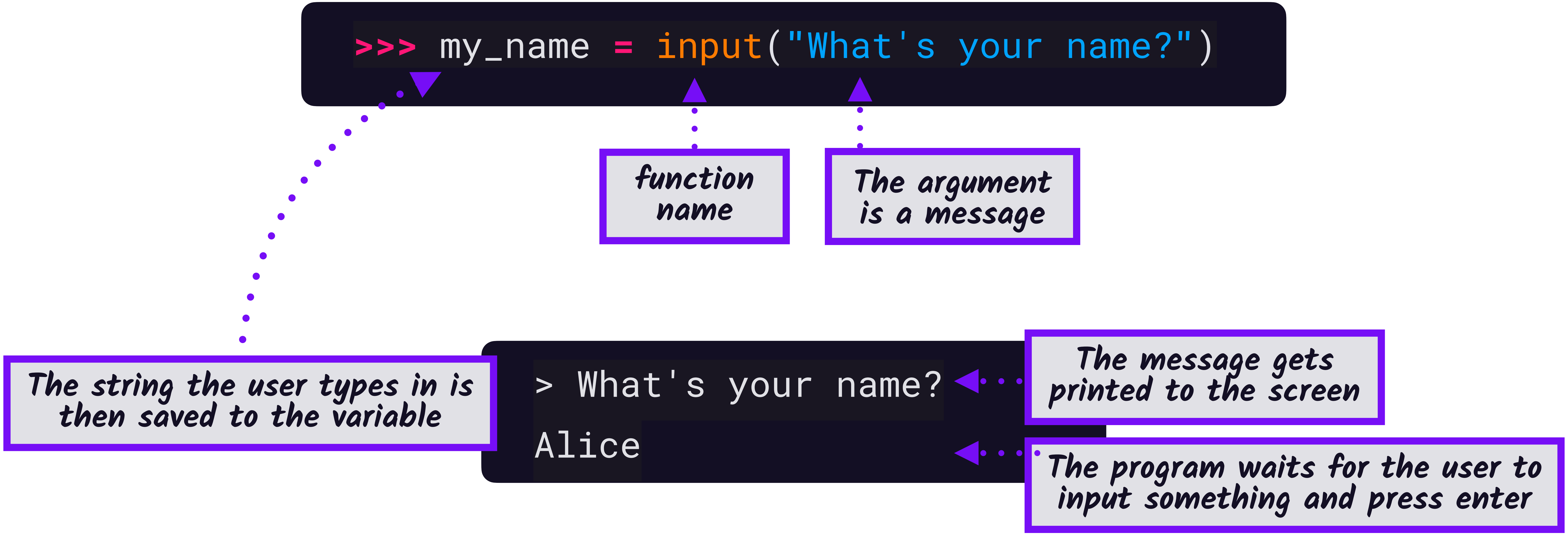
```python
hello = "Hello"
name = "Sarah"
greeting = hello + " " + name
print(greeting)
```

Let's ask the user for their name.

```
> python3 greeting.py
Hello Sarah
```

How can we customize this program for other names?

Python input() Function



```
>>> my_name = input("What's your name?")
```

function name

The argument is a message

```
> What's your name?
Alice
```

The string the user types in is then saved to the variable

The message gets printed to the screen

The program waits for the user to input something and press enter

Console Input

greeting.py

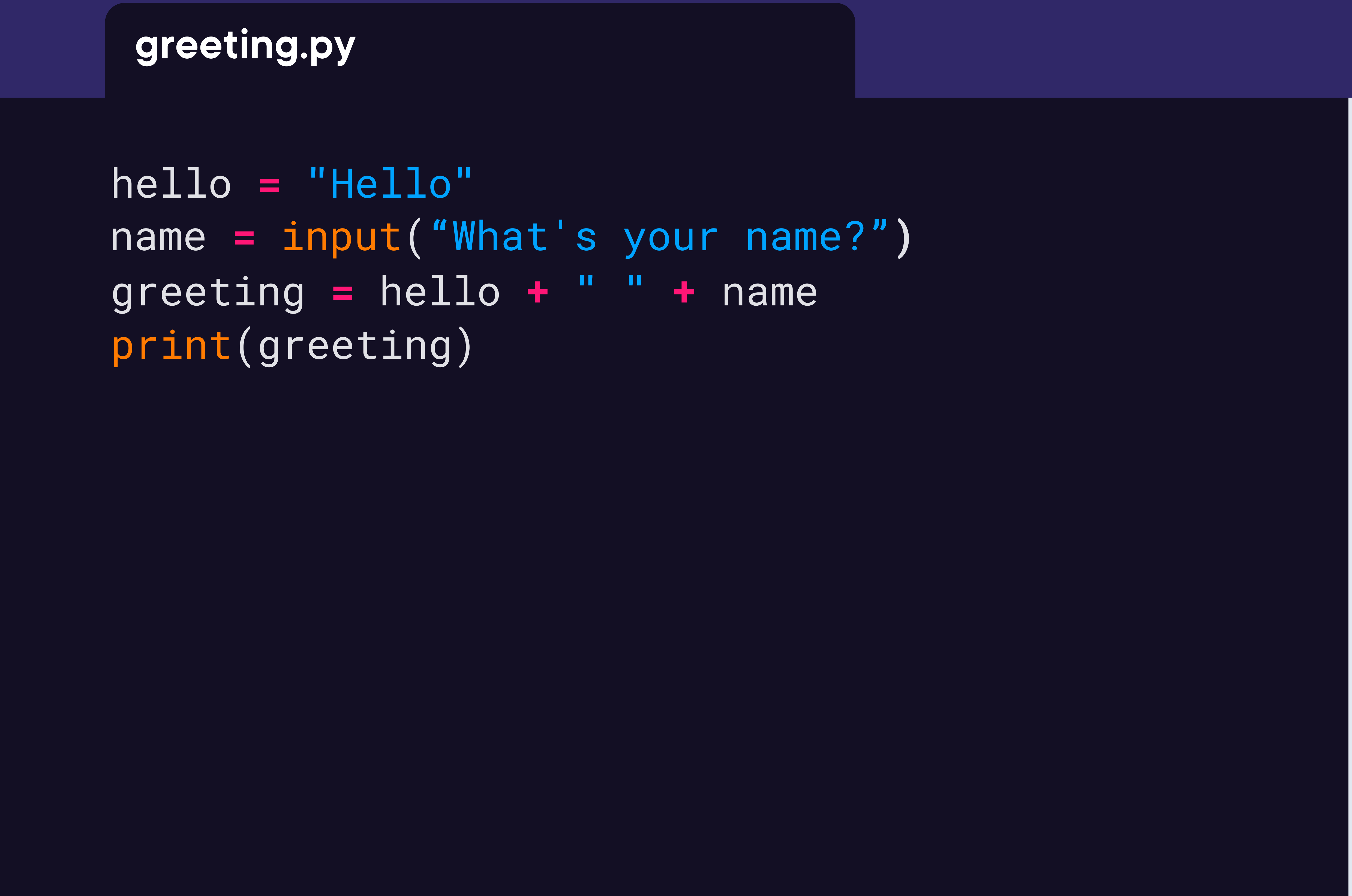
```python
hello = "Hello"
name = input(“What's your name?”)
greeting = hello + " " + name
print(greeting)
```

input() *prints the statement, then waits for a value from the console*

```
> python3 greeting.py
What’s your name?Bob
Hello Bob
```

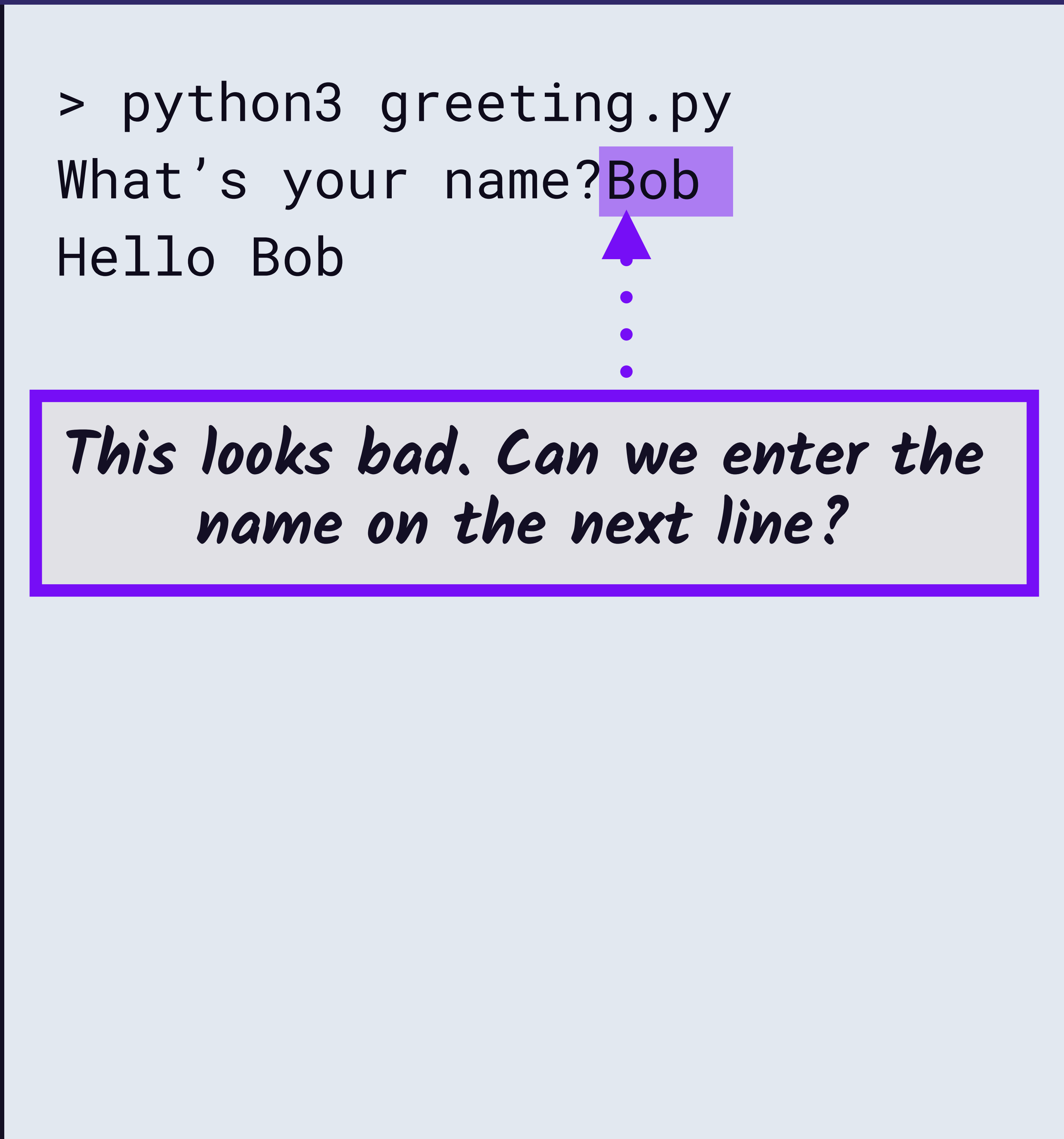
Notice how the name Bob is now printed inside of the greeting.

Console Input



greeting.py

```python
hello = "Hello"
name = input(“What's your name?”)
greeting = hello + " " + name
print(greeting)
```



```
> python3 greeting.py
What’s your name?Bob
Hello Bob
```

This looks bad. Can we enter the name on the next line?

Console Input

greeting.py

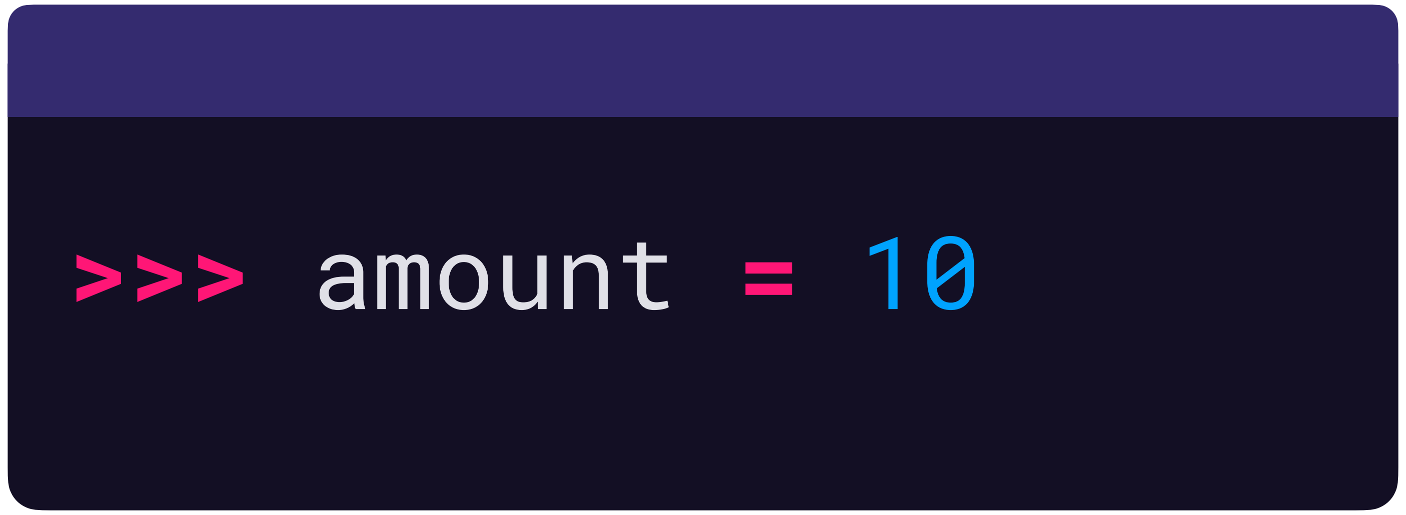
```
hello = "Hello"
name = input(“What's your name?\n”)
greeting = hello + " " + name
print(greeting)
```

\n is a special character for a new line

```
> python3 greeting.py
What’s your name?
Bob
Hello Bob
```

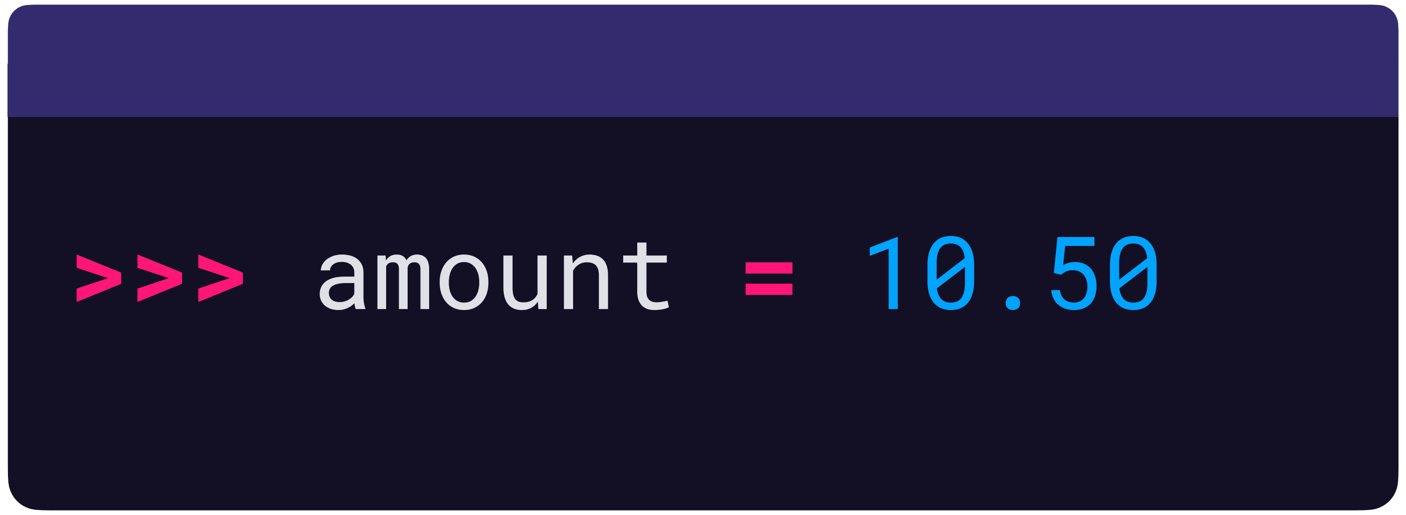
Now input is entered on the next line.

Summary of Primitive Data Types



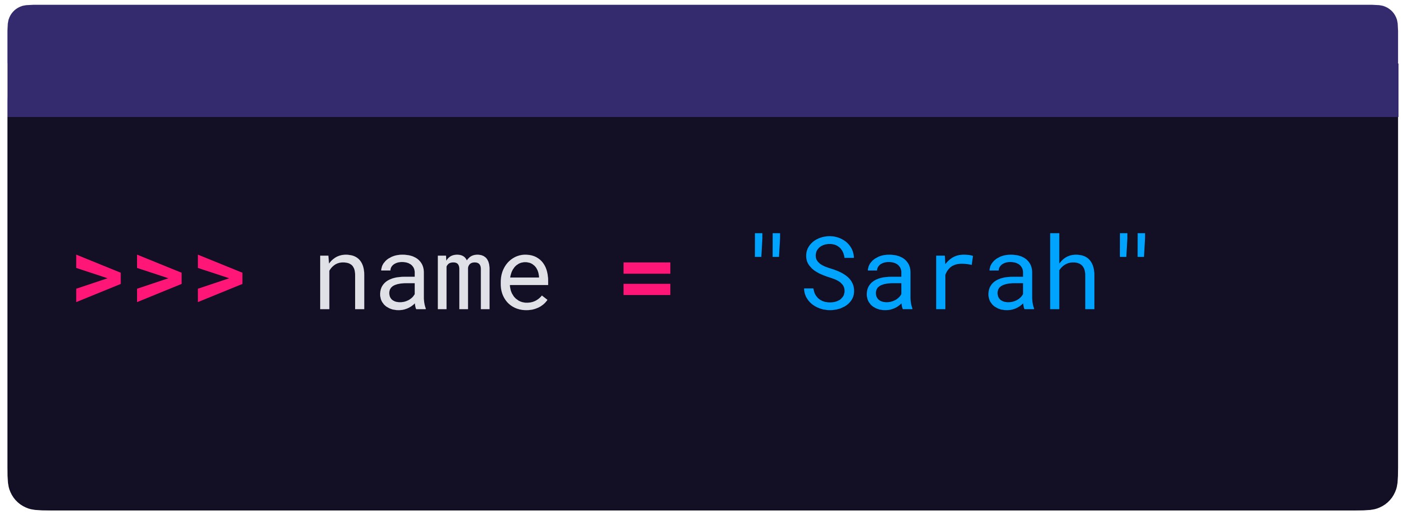
```
>>> amount = 10
```

int



```
>>> amount = 10.50
```

float



```
>>> name = "Sarah"
```

string

Summary of Input and Output

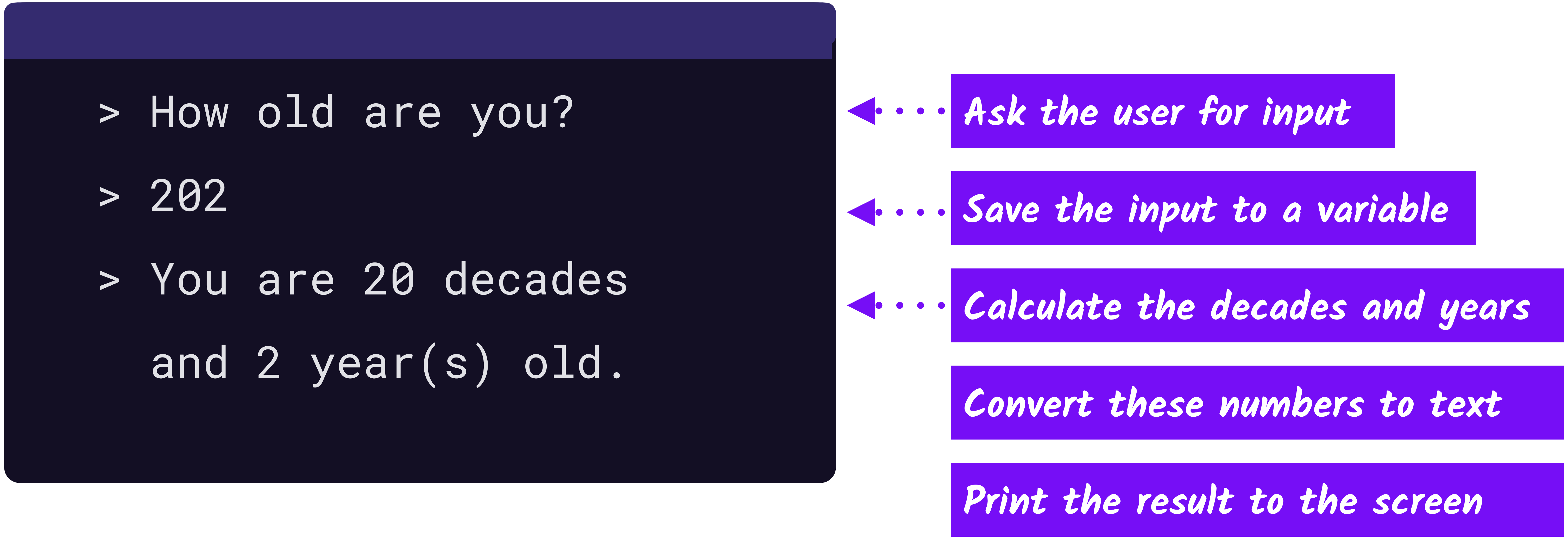
```
>>> name = input(“What's your name?\n”)
What’s your name?
Sarah
```

input

```
>>> print(“Hello ” + name + "!!")
Hello Sarah!!
```

output

Age Calculator



```
> How old are you?
> 202
> You are 20 decades
  and 2 year(s) old.
```

Ask the user for input

Save the input to a variable

Calculate the decades and years

Convert these numbers to text

Print the result to the screen