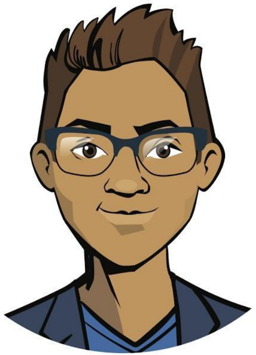
Module 1. Lesson 2. Variables

Confirmation of qualifications

A **confirmation of qualifications** is a regular meeting with a lead programmer.

At this meeting, you will need to demonstrate your professional knowledge and skills.

If your qualifications are confirmed, you will be allowed to work on the project.



Cole
Senior Developer

Confirmation of qualifications

What is an algorithm, a programming language, a program ?

Confirmation of qualifications

What is an algorithm, a programming language, a program ?

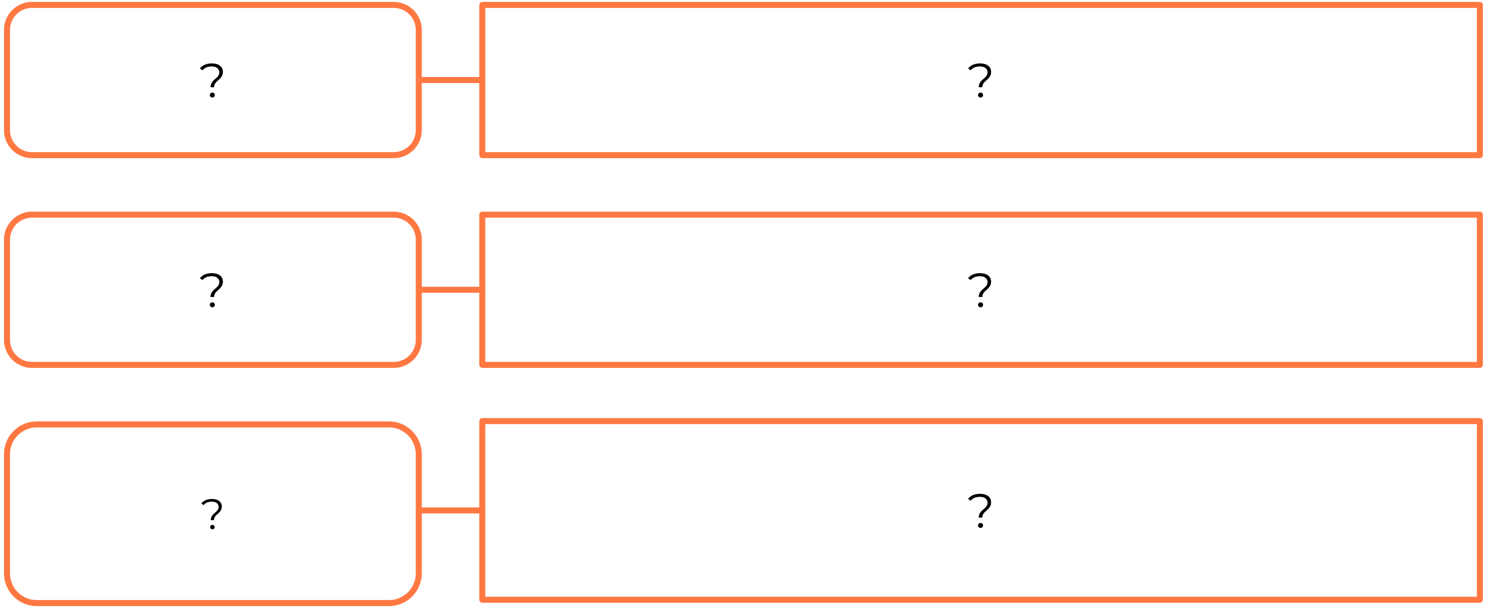
An **algorithm** is a sequence of actions to achieve a goal.

A **programming language** is a language to communicate with machines. It includes commands. Each command has a single meaning.

A **program** is an algorithm written in a programming language.

Confirmation of qualifications

What Python language rules do you know?



?

?

?

?

?

?

Confirmation of qualifications

What Python **language rules** do you know?

Rule of beginning	The first command in a program shall be written in the beginning of a line.
Rule of order	Commands shall be executed in order if they are written one below the other.
Rules of code style	In Python, replacing lowercase letters with uppercase ones is not allowed. An accidental character (even a space or comma) may break the program.

There are other rules, too. Today we will expand this list

Confirmation of qualifications

What is a **function** ? Which function do you already know?

A **function** is...

Confirmation of qualifications

What is a function ? Which function do you already know?

Confirmation of qualifications

A **function** is an algorithm that is composed in a programming language and has a unique name.

`print()` is a function for printing the parameters specified in its parentheses.

Function syntax	*Program will print*
`print(‘Hello, world!’)`	Hello, world!
`print(‘Hello’, ‘world!’)`	Hello, world!

Automatic space

Which parameters can print output?

Confirmation of qualifications

What is specified in the parentheses when we use print?

As arguments, print() may accept words, numbers, and arithmetic expressions.

Function syntax	*Operation*	*Output*
print(2*7)	?	?
print(10+4)	?	?
print(20-6)	?	?
print(28/2)	?	?
print(136%10)	?	?

Confirmation of qualifications

What is specified in the parentheses when we use print?

As arguments, print() may accept words, numbers, and arithmetic expressions.

Function syntax	*Value*	*Output*
`print(2*7)`	Multiplication	14
`print(10+4)`	Sum	14
`print(20-6)`	Difference	14
`print(28/2)`	Quotient	14.0
`print(136%10)`	Division with remainder	6

Confirmation of qualifications

Can we set our own order of operations ?

Confirmation of qualifications

Can we set our own order of operations ?

Yes, we can:

Function syntax	*Output*
`print(1+1*7)`	8
`print((1+1)*7)`	14

In mathematics, division, multiplication, determining the quotient and the remainder are executed first, followed by addition and subtraction.

By using parentheses, we can modify the order of operations.

Confirmation of qualifications

Qualifications confirmed!

Great, you are ready to brainstorm and complete your work task!

Confirmation of qualifications

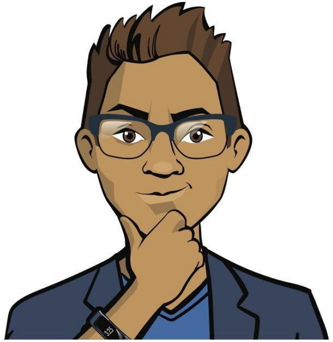
Brainstorm:

Variables

Variables

A **variable** is a data element that has its own name.

Variables are used to work with data that can change.



Brainstorm

Defining a variable

If you want to use a variable in your program, you need to:

- create a variable by giving it a **name**;
- set the variable's **value**.

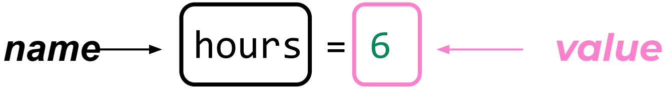
Brainstorm

Defining a variable

If you want to use a variable in your program, you need to:

- create a variable by giving it a **name**;
- set the variable’s **value**.

Example:

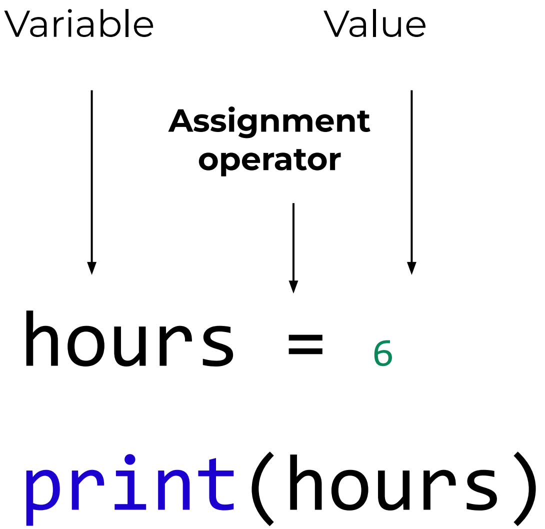


```
hours = 6
minutes = 42
print('According to statistics, in 2019, an Internet user
         spent', hours, 'hours', minutes,
         'minutes online every day')
```

Brainstorm

Assignment operator

To set a variable’s value, you need to use the assignment operator.


Variable

Value

Assignment operator

```
hours = 6

print(hours)
```

The program will print:

```
6
```

Brainstorm

Assignment operator

The assignment operator can <u>change the value</u> of an existing variable.

```
hours = 6

hours = 7.5
```

Changing the initial value

```
print(hours)
```

The program will print:

```
7.5
```

Brainstorm

What name can you give to a variable?

- You **can** use letters, digits, and underscores.
- Every variable’s name **must** begin with a letter or an underscore.
- A variable’s name **must not** use the language’s commands and other reserved names.

Brainstorm

Variable names

Compare some effective and ineffective variable names.

***Ineffective** name*	*Why?*

`a = 56`
`num = 1.34`
	Other programmers will not understand what this variable stands for.

`number_of_students_at_school = 1108`	The name is too long. It is not convenient to use.

Brainstorm

Variable names

Compare some effective and ineffective variable names.

Ineffective *name*	*Why?*

`a = 56`
`num = 1.34`
	Other programmers will not understand what this variable stands for.

`number_of_students_at_school = 1108`	The name is too long. It is not convenient to use.

Effective *name*	*Why?*

`surname = 'Smith'`
`ticket_price = 1999`
`amount_students = 826`
	The name explains the variable's purpose.
	Your fellow programmers will be able to read your program and quickly understand the point.

Brainstorm

Data types

The data hidden behind variable names can be of different types. We know three:

- **integer** numbers,
- **decimal** fractions,
- **string** data.

Different data types can be used to program different actions.

Brainstorm

Data types

The data hidden behind variable names can be of different types. We know three:

- **integer** numbers,
- **decimal** fractions,
- **string** data.

Numeric type		*String type*
144	Integer number (int)	'Calvin' (str)
48.3	Decimal fraction (float)	'256'(str)
(2*11)	Integer number (int)	'15.05.2007' (str)
(4*8.2)	Decimal fraction (float)	'Data received' (str)

Brainstorm

Working with numeric variables

Example. The following program calculates an employee’s salary for a certain period.

What will the program print?

```
daily_salary = 1000

days = 22

total = daily_salary*days

print(total)
```

Brainstorm

Working with numeric variables

Example. The following program calculates an employee’s salary for a certain period.

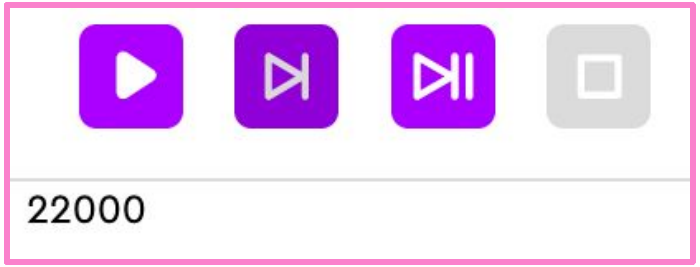
What will the program print?

```
daily_salary = 1000

days = 22

total = daily_salary*days

print(total)
```



```
22000
```

Brainstorm

Working with numeric variables

Example. The following program calculates an employee’s salary for a certain period.

An employer using this program decided to trick one of their employees and **change the daily pay**. What will the program print?

```
daily_salary = 1000

days = 22

daily_salary = 500

total = daily_salary*days

print(total)
```

Brainstorm

Working with numeric variables

Example. The following program calculates an employee's salary for a certain period.

An employer using this program decided to trick one of their employees and **change the daily pay**. What will the program print?

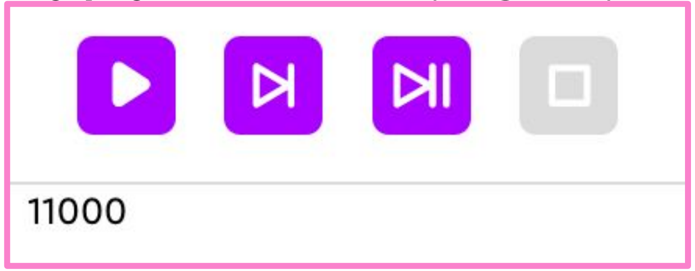
```
daily_salary = 1000

days = 22

daily_salary = 500

total = daily_salary*days

print(total)
```

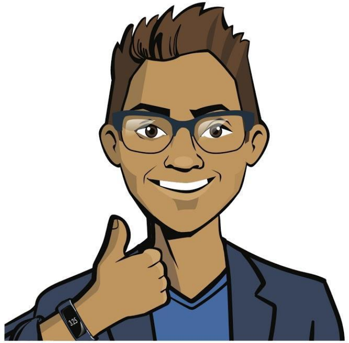


```
11000
```

Brainstorm

Conclusions:

1. A variable is a data element that has its own name.
2. The assignment operator is used to set a variable's initial value or change its current value.
3. The data behind variables can be of different types. For now, we know that there are numeric and string types.

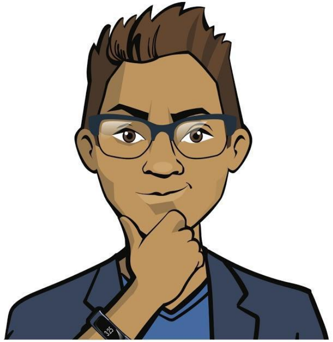


Brainstorm

A broken program

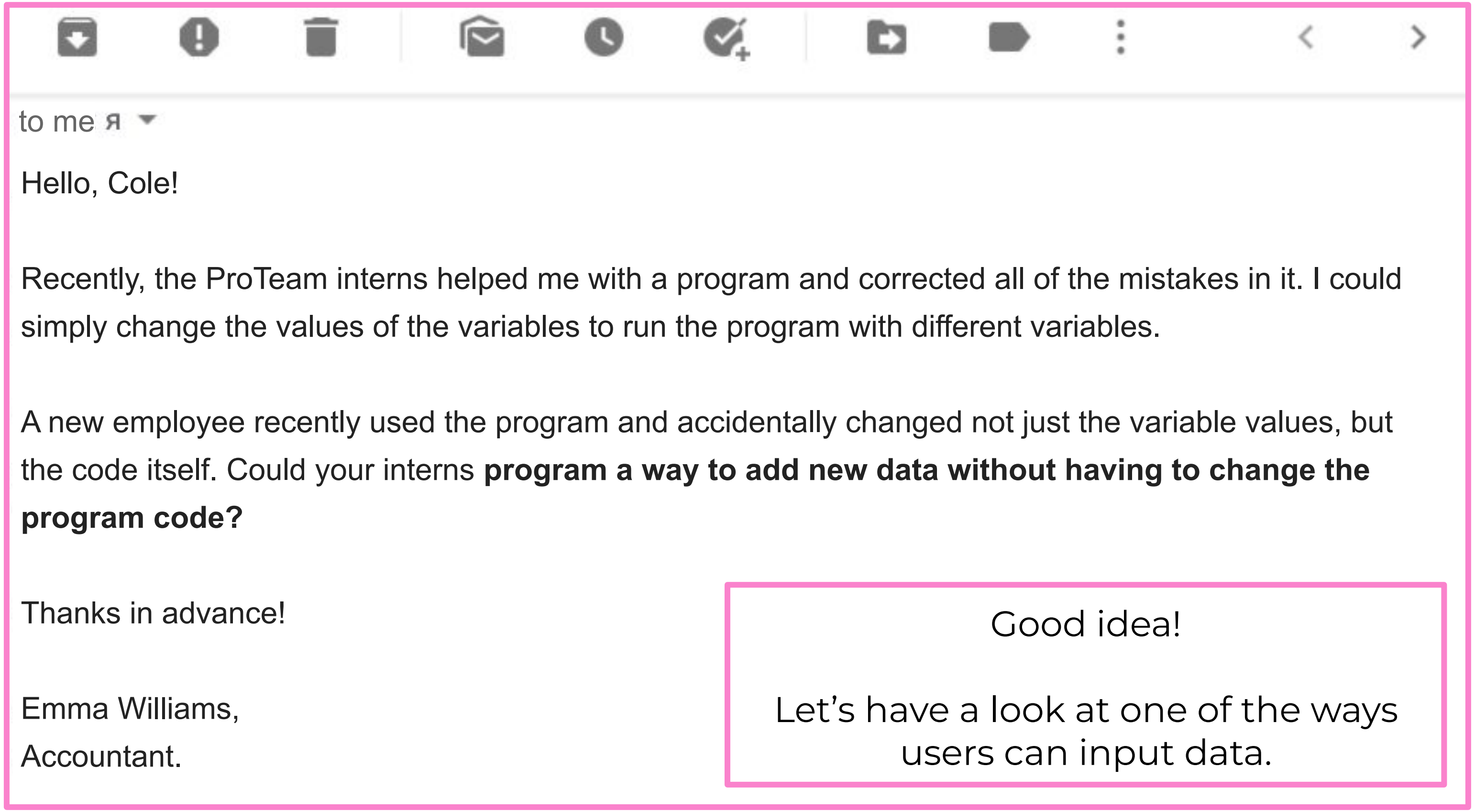
We have just received an email from the accountant of the travel agency you have been working with.

They seem to be concerned about something.



Brainstorm

A broken program



Hello, Cole!

Recently, the ProTeam interns helped me with a program and corrected all of the mistakes in it. I could simply change the values of the variables to run the program with different variables.

A new employee recently used the program and accidentally changed not just the variable values, but the code itself. Could your interns **program a way to add new data without having to change the program code?**

Thanks in advance!

Emma Williams,
Accountant.

Good idea!

Let’s have a look at one of the ways users can input data.

Brainstorm

The input() function

input() is a function to input data from the keyboard.

```
surname = input('Enter the manager’s surname:')
city = input('Enter the city:')
print('The branch manager in', city, ' is', surname)
```

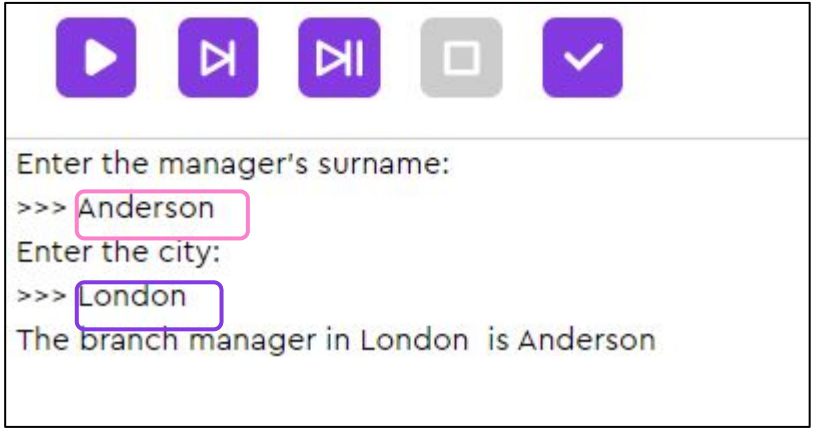
What will the program print?

Brainstorm

The input() function

`input()` is a function to input data from the keyboard.

```
surname = input('Enter the manager’s surname:')
city = input('Enter the city:')
print('The branch manager in', city, ' is', surname)
```



```
Enter the manager's surname:
>>> Anderson
Enter the city:
>>> London
The branch manager in London  is Anderson
```

The program uses user input.

Brainstorm

The input() function

`input()` is a function to input data from the keyboard.

```
result = input('Phrase')
```

The **result** of the algorithm's execution: *a variable.*

The **algorithm** reading the data.

A hint for users.

Brainstorm

The input() function

`input()` is a function to input data from the keyboard.

```
add_services = input('Price of additional services:')

total = 2500 + add_services

print('Total price:', total)
```

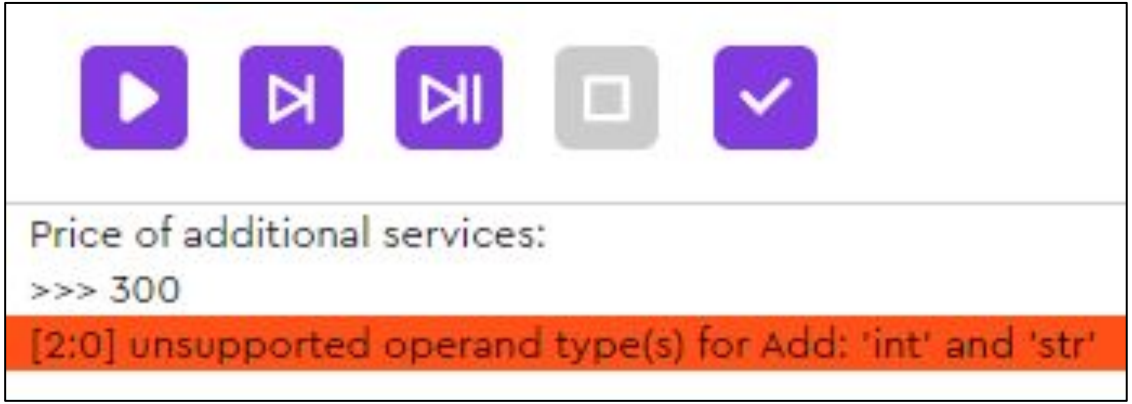
What will the program print?

Brainstorm

The input() function

`input()` is a function to input data from the keyboard.

```
add_services = input('Price of additional services:')
total = 2500 + add_services
print('Total price:', total)
```



```
Price of additional services:
>>> 300
[2:0] unsupported operand type(s) for Add: 'int' and 'str'
```

Looks like there is an error in the program!

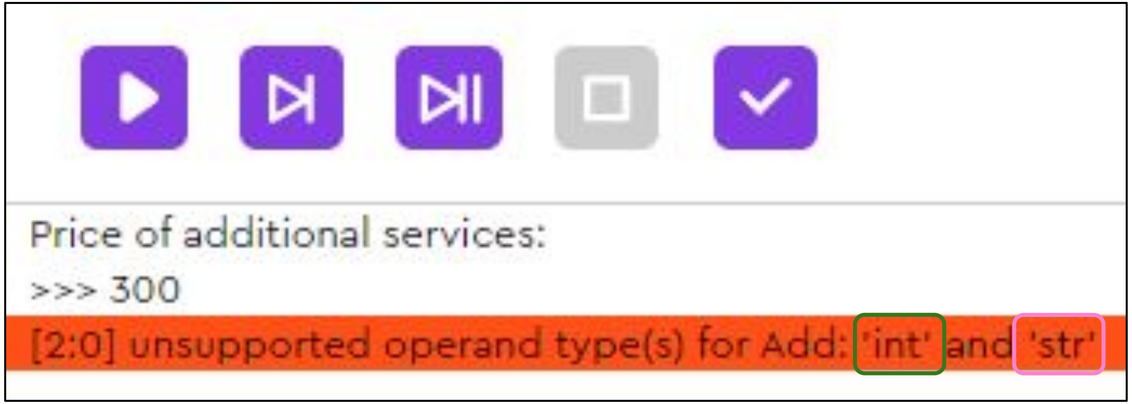
Can you guess what it is?

Brainstorm

The input() function

`input()` is a function to input data from the keyboard.

```
add_services = input('Price of additional services:')
total = 2500 + add_services
print('Total price:', total)
```



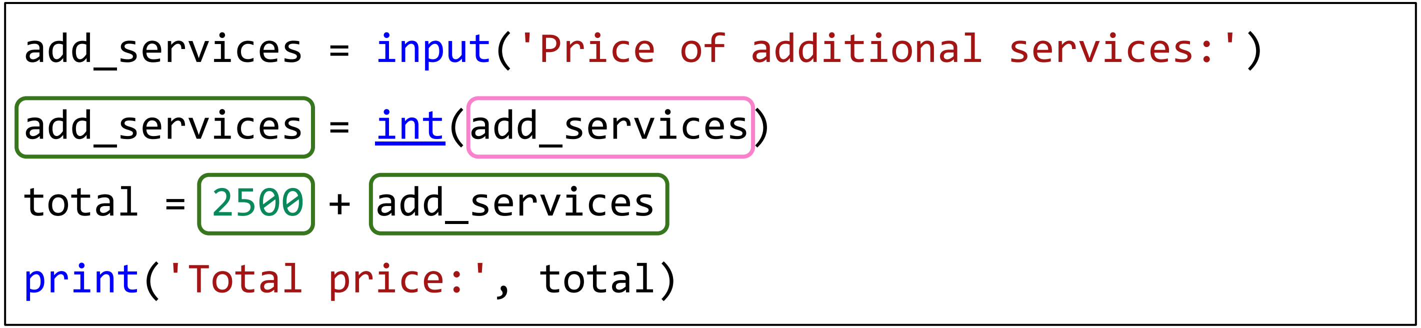
```
Price of additional services:
>>> 300
[2:0] unsupported operand type(s) for Add: 'int' and 'str'
```

The **result** of the `input` function's execution is a **string**, not a number. A computer does not know how to add up integer and string values.

Brainstorm

Int() and str() functions

int() and **str()** functions are used to switch from one data type to another.



```
add_services = input('Price of additional services:')
add_services = int(add_services)
total = 2500 + add_services
print('Total price:', total)
```

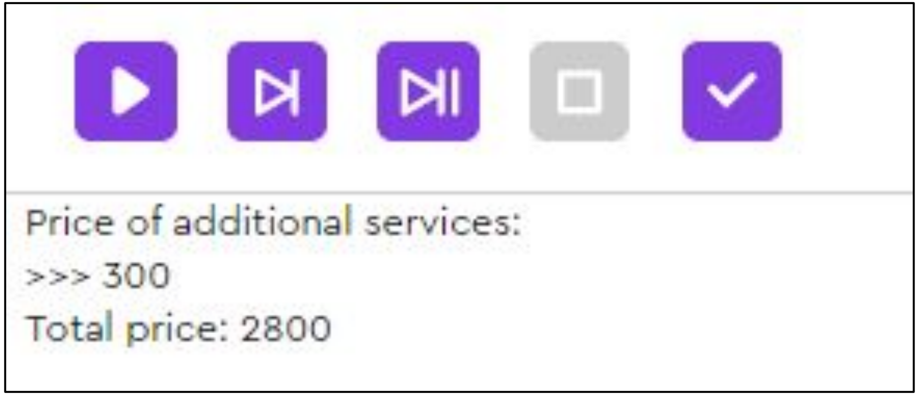
What will the program print?

Brainstorm

Int() and str() functions

`int()` and `str()` functions are used to switch from one data type to another.

```
add_services = input('Price of additional services:')
add_services = int(add_services)
total = 2500 + add_services
print('Total price:', total)
```



```
Price of additional services:
>>> 300
Total price: 2800
```

The computer knows how to add up integer values.

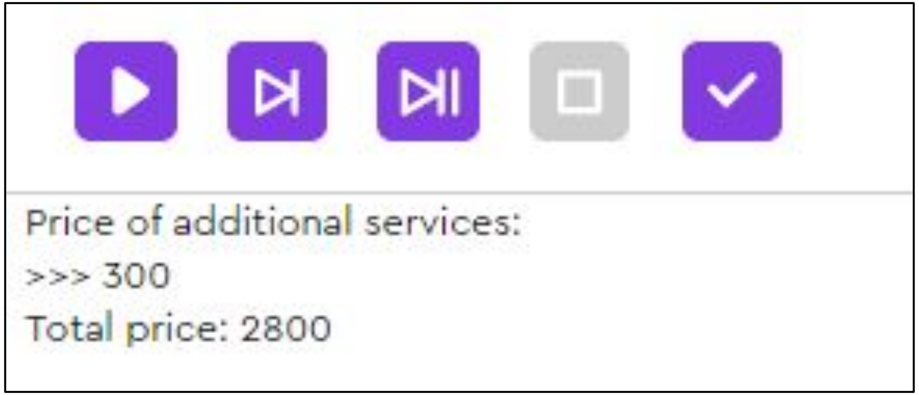
The program works correctly.

Brainstorm

Int() and str() functions

`int()` and `str()` functions are used to switch from one data type to another.

```
add_services = int(input('Price of additional services:'))
total = 2500 + add_services
print('Total price:', total)
```



```
Price of additional services:
>>> 300
Total price: 2800
```

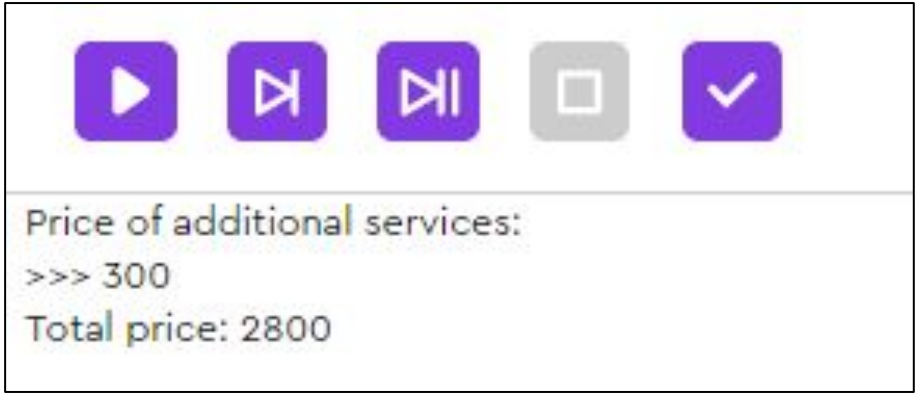
There is a shorter way to write the program.

Brainstorm

Int() and str() functions

int() and str() functions are used to switch from one data type to another.

```
add_services = input('Price of additional services:')
add_services = int(add_services)
total = 2500 + add_services
print('Total price:', total)
```



```
Price of additional services:
>>> 300
Total price: 2800
```

The str() function switches data to the string data type.

We will discuss it in detail next time.

Brainstorm