Confirmation of qualifications

To get started on today's tasks, demonstrate **your knowledge level**.

Prove that you are ready for the brainstorm!

What is a **variable** ?
How is it created?

What is a variable ? How is it created?

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

To use a variable in a program, you need to:

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

Example:

hours = 6

Which operator can **change** the value of a variable?

Which operator can **change** the value of a variable?

The **assignment operator** can change the value of an existing variable.

```
hours = 6

hours = 7.5
```

Changing the initial value

```
print(hours)
```

The program will print:
7.5

Choose the good variable names. Explain your choice.

a

encryption

username

www

total_amount_of_students

book_title

pr

Confirmation of qualifications

Choose the good variable names. Explain your choice.

Meaning is unclear

a

Name is unreadable

encryption

Meaning is unclear

www

username

Name is too long

total_amount_of_students

book_title

Meaning is unclear

pr

Confirmation of qualifications

Which **data types** do you know?

Which **data types** do you know?

We know three:

- **integer** numbers,
- **decimal** fractions,
- **strings** .

Numbers		Strings
144	<u>Integer</u> number (int)	`'John'` (str)
48.3	Decimal fraction (float)	`'256'`(str)
(2*11)	<u>Integer</u> number (int)	`'15.05.2007'` (str)
(4*8.2)	Decimal fraction (float)	`'Data received'` (str)

Confirmation of qualifications

Which **function** is used to **input data** from the keyboard?

Which **function** is used to **input data** from the keyboard?

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

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

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

The name of the function that reads the data.

A hint for users.

Which functions are used to **switch from one data type to another**? When are they used?

Which functions are used to switch from one data type to another ? When are they used?

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

```python
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 **result** of the input function's execution is a **string**, not a number.

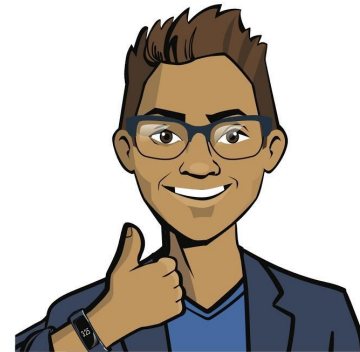We need to switch from the string type to the integer.

Qualifications confirmed!

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

Brainstorm:

Strings

String variables

To learn to extract information from strings, let's begin by programming some simple cases for the "Sunflower" facility.

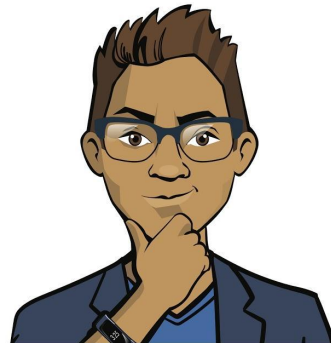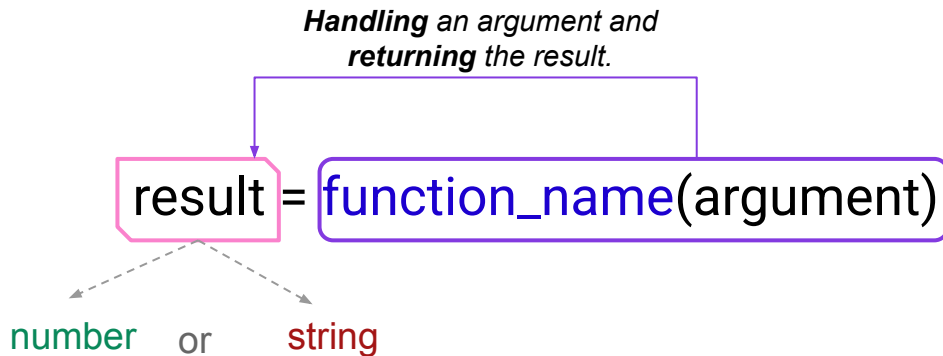| *Query*	*Required skills*
The manager offers a discount for large comments.	**Determine the length** of a string.
The gym part of the comment may be saved for advertising	**Cut a part out of** a string.
The chef is interested in mentions of the summer menu.	**Search for** a word or phrase in a string.
The manager appreciates comments containing the word "cozy"	**Search for** a word or phrase in a string.

Brainstorm

String variables

All the functions we will look at later **return** a **value** of some kind as a result.

This makes sense, since we want to extract some information from a string. This information can be saved in a dedicated variable.

Handling an argument and returning the result.

result = function_name(argument)

number or string

Operations on string variables

Task 1: Customers who post large comments are rewarded with discounts. Write a program that prints the length of a comment:

"A fantastic place!"

```
length = len(string)
```
——— this function obtains the length of a string provided in parentheses. The returned value is the number of characters in the string.

Program	Program will print
feedback = 'A fantastic place!' ?	18

Operations on string variables

Task 1: Customers who post large comments are rewarded with discounts. Write a program that prints the length of a comment:

"A fantastic place!"

```
length = len(string)
```
—— this function obtains the length of a string provided in parentheses. The returned value is the number of characters in the string.

Program	Program will print
feedback = 'A fantastic place!' length = len(feedback) print(length)	18

Operations on string variables

Task 2: The facility manager has asked us to cut out and save the part about the gym for advertising. Write a program that prints the required part of the comment:

"_Went for boxing_. A good beach"

How can we print at least one character of the comment?

Operations on string variables

Task 2: The facility manager has asked us to cut out and save the part about the gym for advertising. Write a program that prints the required part of the comment:

"Went for boxing. A good beach"

W	e	n	t		f	o	r		b	o	x	i	n	g	.
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

...

Python enumerates characters starting from zero.

In Python, the string is an object consisting of multiple elements – characters.

Brainstorm

Operations on string variables

Task 2: The facility manager has asked us to cut out and save the part about the gym for advertising. Write a program that prints the required part of the comment:

"Went for boxing. A good beach"

W	e	n	t		f	o	r		b	o	x	i	n	g	.
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

...

```
symbol = feedback[0]
```
—— this is a tool that obtains a single character based on its number.
A character's number is specified in [].

Brainstorm

Operations on string variables

Task 2: The facility manager has asked us to cut out and save the part about the gym for advertising. Write a program that prints the required part of the comment:

"Went for boxing. A good beach"

W	e	n	t		f	o	r		b	o	x	i	n	g	.
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

...

Program	Program will print
feedback = 'Went for boxing. A good beach' symbol = feedback[0] print(symbol)	W

Brainstorm

Operations on string variables

Task 2: The facility manager has asked us to cut out and save the part about the gym for advertising. Write a program that prints the required part of the comment:

"Went for boxing. A good beach"

W	e	n	t		f	o	r		b	o	x	i	n	g	.
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

A		g	o	o	d		b	e	a	c	h
16	17	18	19	20	21	22	23	24	25	26	27

To obtain a part of a substring, you need to know the start and end numbers of the required part.

Operations on string variables

Task 2: The facility manager has asked us to cut out and save the part about the gym for advertising. Write a program that prints the required part of the comment:

"**_W_**_ent for boxin**g**. A good beach_"

w	e	n	t		f	o	r		b	o	x	i	n	g	.
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

...

`feedback[0:15]` – you can get a part of the string by specifying in square brackets the number of the beginning and the number following the end of the substring.

Numbers are separated by a colon.

Operations on string variables

Task 2: The facility manager has asked us to cut out and save the part about the gym for advertising. Write a program that prints the required part of the comment:

"Went for boxing. A good beach"

w	e	n	t		f	o	r		b	o	x	i	n	g	.
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

…

Program	*Program will print*

`feedback = 'Went for boxing. A good beach'`
`sport = feedback[0:15]`
`print(sport)`
	Went for boxing

Operations on string variables

Task 3a : The chef wants to know if their summer menu is mentioned in the comments. Write a program that determines if a comment contains the word "paella":

"We liked Chef's paella"

```
position = feedback.find('paella')
```

is a function for searching a string for a word specified in parentheses.

Operations on string variables

Task 3a : The chef wants to know if their summer menu is mentioned in the comments. Write a program that determines if a comment contains the word "paella":

"We liked Chef's paella"

```
position = feedback.find('paella')
```

Number of the first character in "paella".

is a function for searching a string for a word specified in parentheses.

W	e		l	i	k	e	d		C	h	e	f	'	s		p	a	e
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18

...

Operations on string variables

Task 3a : The chef wants to know if their summer menu is mentioned in the comments. Write a program that determines if a comment contains the word "paella":

"We liked Chef's paella"

```
position = feedback.find('paella')
```

is a function for searching a string for a word specified in parentheses.

Program	*Program will print*
feedback = 'We liked Chef's paella' position = feedback.find('paella') print(position)	16

Operations on string variables

Task 3b : The chef wants to know if their summer menu is mentioned in the comments. Write a program that determines if a comment contains the word "paella":

"We liked the pizza with pineapples"

```
position = feedback.find('paella')
```
is a function for searching a string for a word specified in parentheses.

Program	Program will print
`feedback = 'We liked the pizza with pineapples'` `position = feedback.find('paella')` `print(position)`	`-1` *This word is absent.*

Operations on string variables

Task 4. The manager also requested a search for comments containing the word "cozy". When leaving, they noticed that some visitors' comments contain the single word "cozy".

"Cozy", "hotel's cozy"

Program	Program will print

```feedback1 = 'Cozy!'```
 	
```feedback2 = 'Hotel's cozy'```
 	
```position1 = feedback1.find('cozy')```
 	
```position2 = feedback2.find('cozy')```
 	
```print(position1)```
 	
```print(position2)```
	What will the program print?

Operations on string variables

Task 4. The manager also requested a search for comments containing the word "cozy". When leaving, they noticed that some visitors' comments contain the single word "cozy".

"Cozy", "hotel's cozy"

| Program	Program will print
feedback1 = 'Cozy!'	-1
feedback2 = "Hotel's cozy"	8
position1 = feedback1.find('cozy')	
position2 = feedback2.find('cozy')	
print(position1)	
print(position2)	*Why?!*

Operations on string variables

Task 4. The manager also requested a search for comments containing the word "cozy". When leaving, they noticed that some visitors' comments contain the single word "cozy".

"Cozy", "hotel's cozy"

Compare:

C	o	z	y		!
0	1	2	3	4	5

H	o	t	e	l	'	s		c	o	z	y
0	1	2	3	4	5	6	7	8	9	10	11

Brainstorm

Operations on string variables

Task 4. The manager also requested a search for comments containing the word "cozy". When leaving, they noticed that some visitors' comments contain the single word "cozy".

"Cozy", "hotel's cozy"

Compare:

C	o	z	y	_	!
0	1	2	3	4	5

H	o	t	e	l	'	s		c	o	z	y
0	1	2	3	4	5	6	7	8	9	10	11

Uppercase letter

Lowercase letter

Right! We know from the code style rules that the case matters! What could help us with this issue?

Brainstorm

Operations on string variables

Task 4. The manager also requested a search for comments containing the word "cozy". When leaving, they noticed that some visitors' comments contain the single word "cozy".

"Cozy", "hotel's cozy"

```
feedback = feedback.lower()
```
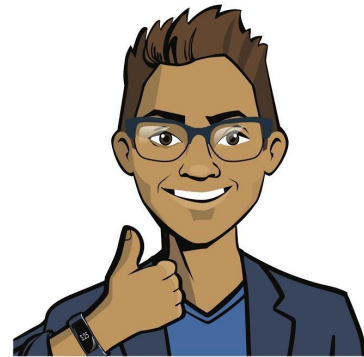— this function converts all the letters in a string into lowercase.

C	o	z	y	_	!
0	1	2	3	4	5
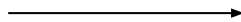
→

c	o	z	y	_	!
0	1	2	3	4	5

Operations on string variables

Task 4. The manager also requested a search for comments containing the word "cozy". When leaving, they noticed that some visitors' comments contain the single word "cozy".

"Cozy", "hotel's cozy"

`feedback = feedback.lower()` — this function converts all the letters in a string into lowercase.

Program	*Program will print*

`feedback1 = 'Cozy!'`
`feedback1 = feedback1.lower()`
`position1 = feedback1.find('cozy')`
`print(position1)`
	`0`

Conclusions:

1. String variables allow a number of operations, from calculating a string's length to searching for a word in a string.
2. All the characters in a string are numbered. The first character's number is 0.
3. To master those functions, you need to practice them!

"Sunflowers":
Feedback

Weird program

A junior developer has written a program to calculate a service's quality rating. Is it absolutely correct?

```python
point1 = input('Rate the hotel's convenience from 1 to 5:')

point2 = input('Rate the restaurant's food from 1 to 5:')

total_rating = point1 + point2

print('Total rating:', total_rating)
```

Brainstorm

Weird program

A junior developer has written a program to calculate a service's quality rating. Is it absolutely correct?

```
point1 = input('Rate the hotel's convenience from 1 to 5:')
point2 = input('Rate the restaurant's food from 1 to 5:')
total_rating = point1 + point2
print('Total rating:', total_rating)
```

The junior developer forgot to switch from the string type to integer using int()!

The program will generate an error!

Brainstorm

Weird program

A junior developer has written a program to calculate a service's quality rating. Is it absolutely correct?

```python
point1 = input('Rate the hotel's convenience from 1 to 5:')

point2 = input('Rate the restaurant food from 1 to 5:')

total_rating = point1 + point2

print('Total rating:', total_rating)
```

Rate the hotel's convenience 1 to 5:
>>> 3
Rate the restaraunt food 1 to 5:
>>> 4
Total rating: 34

Weird, the program works...

*Where did the number **34** come from?*

Weird program

The computer has recognized the entered data as strings.

Since strings cannot be added to one another, the computer merged them!

```python
point1 = input('Rate the hotel's convenience from 1 to 5:')

point2 = input('Rate the restaurant food from 1 to 5:')

total_rating = point1 + point2

print('Total rating:', total_rating)
```



```
Rate the hotel's convenience 1 to 5:
>>> 3
Rate the restaraunt food 1 to 5:
>>> 4
Total rating: 34
```
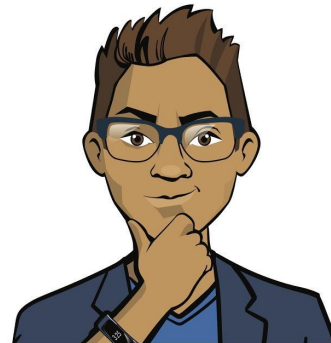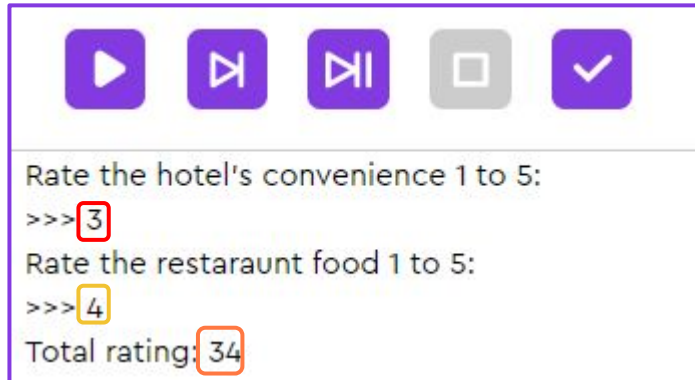
Weird program

Junior developers, you should figure out how a computer interprets programs so these simple things do not give you a nasty surprise!

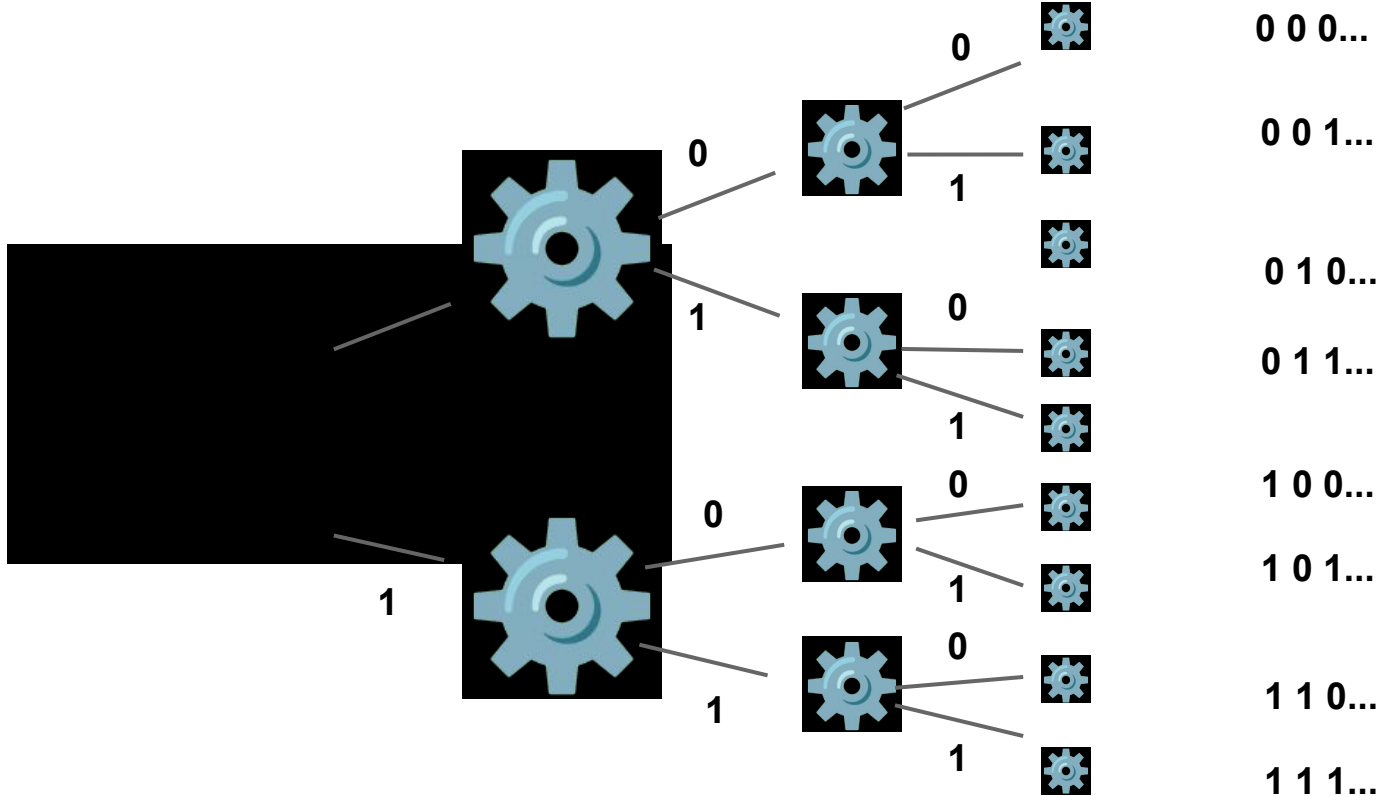This will help you understand why + can add numbers and concatenate strings.
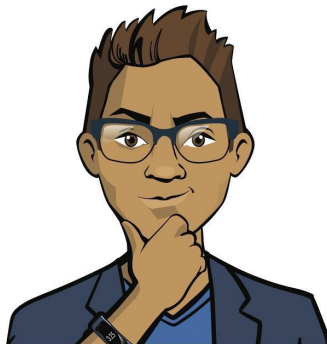
Computers only understand electric signals

1 — signal is present
0 — signal is absent

Computer consists of numerous parts. Each one can recognize signals.

0 0 0...

0 0 1...

0 1 0...

0 1 1...

1 0 0...

1 0 1...

1 1 0...

1 1 1...

To get a computer to perform an action, you need to submit a command consisting of 0s and 1s.

0 0 0... —— *will perform* ——→ *Action 1*

0 0 1... —— *will perform* ——→ *Action 2*

0 1 0... —— *will perform* ——→ *Action 3*

"Programming" a computer with these kinds of signals is very inconvenient! No one can remember commands composed of 0 and 1.
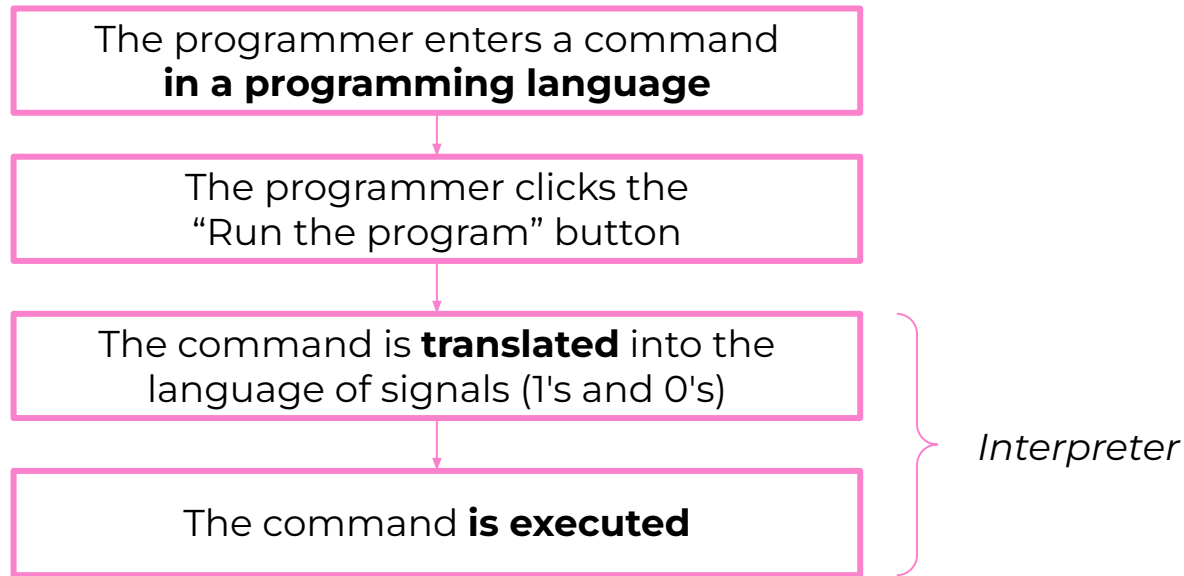
Brainstorm

Programming languages and interpreters

One solution to this problem is to use a programming language and an **interpreter.**

The programmer enters a command **in a programming language**

↓

The programmer clicks the "Run the program" button

↓

The command is **translated** into the language of signals (1's and 0's)

↓

The command **is executed**

Interpreter

An interpreter

is a special program that recognizes and executes commands.

An interpreter

is a special program that recognizes and executes commands.

Let's go back to the example with data types:

```
point1 = input('Rate the hotel's convenience from 1 to 5:')

point2 = input('Rate the restaurant's food from 1 to 5:')

total_rating = point1 + point2
```

Recognition: there is an operator between the strings.

Command: concatenate the strings.

An interpreter
is a special program that recognizes and executes commands.

Consider another example:

```
point1 = 4

point2 = 5

total_rating = point1 + point2
```

Recognition: there is an operator between the numbers.

Command: add the two numbers.

Brainstorm

An interpreter

is a special program that recognizes and executes commands.

The interpreter can be told explicitly which data type it is dealing with. You already know these functions: `int`() and `str`().

```python
point1 = input('Rate the hotel's convenience from 1 to 5:')
point1 = int(point1)
point2 = input('Rate the hotel's convenience from 1 to 5:')
point2 = int(point2)
total_rating = point1 + point2
```

Recognition: there is an operator between the numbers.

Command: add the two numbers.

An interpreter
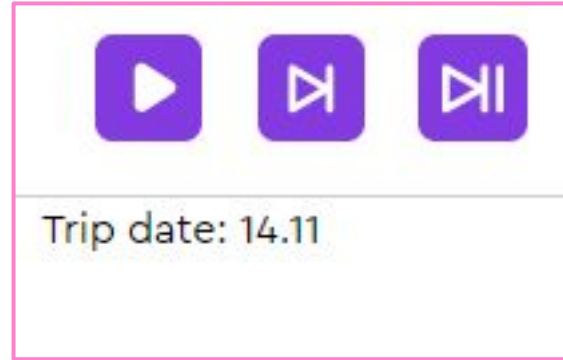## is a special program that recognizes and executes commands.

Programmers tend to call `int()` and `str()` datatype switching functions.

```
day = 14
month = 11
day = str(day)
month = str(month)
date = day + '.' + month
print('Trip date:', date)
```

Recognition: there is an operator between the strings.

Command: concatenate the strings.

Trip date: 14.11

Thanks to the smart interpreter, some operators handle different types differently.

Operator	Meaning for strings	Meaning for numbers
+	Concatenation of strings	The sum of the numbers
*	Multiple repetitions of a string	Multiplication of numbers

Brainstorm

Thanks to the smart interpreter, some operators handle different types differently.

Operator	Meaning for strings	Meaning for numbers
+	Concatenation of strings	The sum of the numbers
*	Multiple repetitions of a string	Multiplication of numbers

'Great' + ' place'	Great place	Concatenation of two strings
3 * 'Cool '	Cool Cool Cool	Repetition of a string 3 times
'Great' * ' place'	can't multiply sequence by non-int of type 'str'	The interpreter does not understand how many times to repeat the string

Brainstorm

Thanks to the smart interpreter, some operators handle different types differently.

Operator	Meaning for strings	Meaning for numbers
+	Concatenation of strings	The sum of the numbers
*	Multiple repetitions of a string	Multiplication of numbers

Also, keep in mind that many functions and operators only work with certain types.
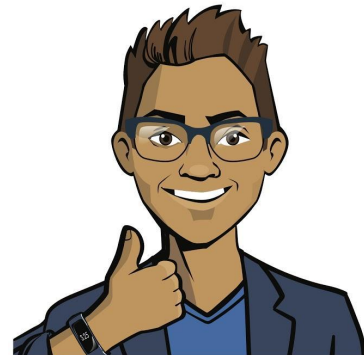
The len() function cannot be used to determine a number's length.

Brainstorm

Conclusions:

1. **The interpreter** is a special program to recognize and execute commands.

2. In recognition, the interpreter <u>focuses on the types of data</u> it interacts with.

3. When necessary, we can change data types using int() and str().