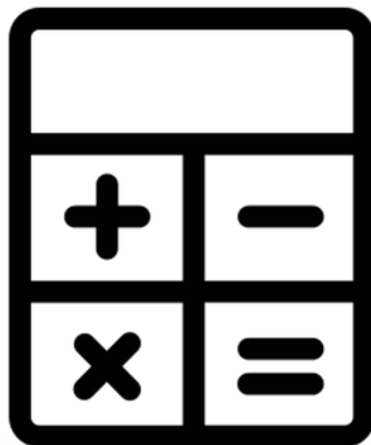


Computer Science Programming Fundamentals

Calculations



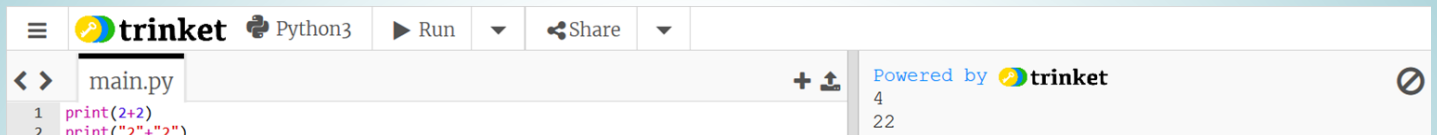
💡 Calculations Intro

Computers are great at math problems! How can we tell Python to solve a math problem for us?

In this lesson, we'll learn about how to use numbers in Python and the special symbols we use to tell it what kind of **calculation** to do.

Calculations pt 1

- Run the code in the example.



The screenshot shows a Trinket Python3 editor interface. The code in the editor is:

```
1 print(2+2)
2 print("2"+"2")
```

The output on the right shows the results of the execution:

```
4
22
```

The first line, `print(2+2)`, correctly outputs `4`. The second line, `print("2"+"2")`, outputs `22` because it concatenates the two strings "2" and "2".

Q: Why does line two give the wrong Answer?

A: When we do math in Python, we can't use strings. We have to use numbers. The first line uses two numbers. Both of them are **integers** (called `int` in Python).

- Run each example:

Subtraction:

```
print(2 - 2)
```

Multiplication:

```
print(2 * 2)
```

Division:

```
print(2 / 2)
```

Calculations pt 2

Q: Why did the last statement output 1.0?

A: When Python does division, it uses a different kind of number called a **float**. Floats always have a decimal point. Integers are always whole numbers and do not have decimal points.

- Run:

```
print(7/2)
```

Calculations in Python follow the Order of Operations, which is sometimes called **PEMDAS**.

- Run:

```
print((6 - 2) * 5)
```

```
print(6 - 2 * 5)
```

Calculations pt 3

The first statement is **evaluated** by Python like this:

1. $(6 - 2) * 5$ *Parentheses first*
2. $4 * 5$
3. 20

The second statement is evaluated like this:

1. $6 - 2 * 5$ *Multiplication first*
2. $6 - 10$
3. -4

Calculations pt 4

The **modulo** operator (%) finds the remainder of the first number divided by the second number.

- Run:

```
print(12 % 10)
```

12 / 10 = 1 with a remainder of 2.

Integer division (//) is like normal division, but it rounds down if there is a decimal point.

- Run:

```
print(5 // 2)
```

5 / 2 = 2.5, which is rounded down to 2

Exponentiation (**) raises the first number to the power of the second number.

- Run:

```
print(3 ** 2)
```

This is the same as 3²

Calculations pt 5

Remember, `input()` **returns** a string, and we can't do math with strings. Fortunately, we can change strings into ints like so:

```
two = "2"
```

```
two = int(two)
```

```
print(two + two)
```

If you need to work with a decimal point, you can change it to a float instead:

```
two = float(two)
```

Calculations pt 5



Activity 1:

Do you know anyone who tends to one-up you in conversation? In this activity, we'll make a simple chatbot that asks a series of questions, explaining to you why it's superior after each one. The robot will have a variable level of one-upmanship.

We'll use `print`, `input`, and math operators and variables to accomplish this.

Calculations pt 6

Example:

```
one_up_level = 1
a1 = input("How many seconds does it take you to
run the 100 meter dash?")
a1 = int(a1)
print("That's cool. I can do it in", a1 -
one_up_level, "seconds though. And I don't even
have legs, sooooo...")
a2 = input("But what about your GPA? I'm sure
that's pretty good, eh? (Enter your GPA)")
a2 = float(a2)
print("Alright. Mine was", a2 + one_up_level)
print("Not that it matters, lol")
```



Explanation:

No matter what number you tell the chatbot, in his calculations it'll always inform you that he's somehow better. It does this by adding or subtracting, depending on which operation flatters it. Many students delete this program after the exercise.

Calculations pt 7



Activity 2:

Make a simple program that tells you if a given number is a multiple of another given number.

A Correct Answer:

```
print("Is _ a multiple of _?")
num1 = input("Write the first number: ")
num2 = input("Write the second number: ")
print(int(num1) % int(num2))
print("If the number above is zero, then", num1,
"is a multiple of", num2)
```



Explanation:

Remember, the modulo finds the remainder of the first number divided by the second number. If it gives us zero, then we know that the second number divides evenly into the first number.