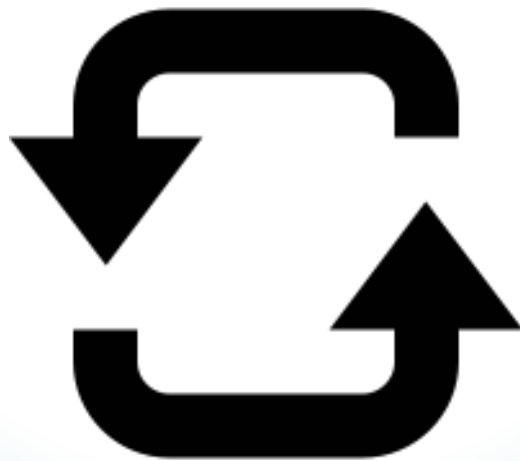


Computer Science Programming Fundamentals

Loops (For)



Loops (For) Intro

Imagine buying new shoes, wearing them once, then putting them in your closet and never touching them again. Instead of wearing them again, you buy another pair. What happens if you do this every day?

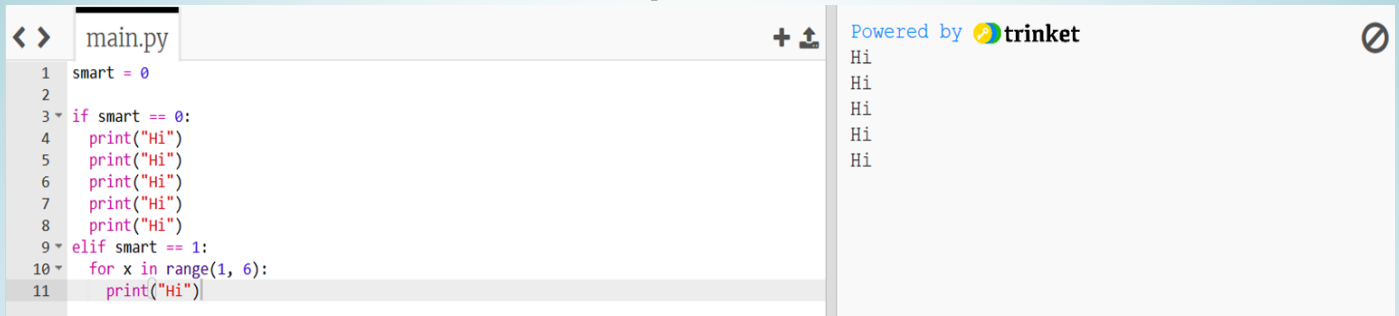
Very soon, your closet will be full of shoes, and you will waste a LOT of money!

When we write code, we don't want to spend extra time writing the same things over and over again. If we already have code in our program that prints "Hi", we should avoid writing another line that also prints "Hi."

Loops are one way to reuse pieces of code. Let's learn about for loops.

Loops (For) pt 1

- Run the example:



```
1 smart = 0
2
3 if smart == 0:
4     print("Hi")
5     print("Hi")
6     print("Hi")
7     print("Hi")
8     print("Hi")
9 elif smart == 1:
10     for x in range(1, 6):
11         print("Hi")
```

Powered by trinket

Hi
Hi
Hi
Hi
Hi

- Change the value of `smart` to 1, then run it again.

The output is the same, but the code making that output is different.

When `smart` is set to 0, Python runs five different print statements.

When `smart` is set to 1, Python uses a for loop to print "Hi" 5 times.

Imagine how much time this would save if you had to print something 1,000 times!

This is a beautiful thing called **iteration**.

Loops (For) pt 2

For tells Python that you want to make a for loop. But what does `x` do?

- Change:

```
for x in range(1, 6):  
    print("Hi")
```

to

```
for x in range(1, 6):  
    print(x)
```

- Make sure `smart` is still set to 1, then run it.

Q: Based on the output, what is the value of `x`?

A: The value of `x` changes each time the for loop runs. The first time the loop runs, it is 0. The next time, it's one. Then two, then three, and so on.

! **Note:** `x` only works inside the for loop!

Loops (For) pt 3

Q: Why does the loop run 5 times?

A: In this case, the `range` function tells the for loop how many times to run. We gave the function two numbers: a bottom (1), and a top (6).

The loop starts on the bottom number and stops right before the top number. If you want to go to 20, your top should be 21. If you want to go to 100, your top should be 101.

Loops (For) pt 4



Activity:

- We'll make a game where two cars race. [Link](#)

```
import random
```

```
import time
```

```
car = "o~o"
```

```
space = " "
```

```
roadside = "-----"
```

```
distance1 = 0
```

```
distance2 = 0
```

```
speed = 3
```

```
for x in range(1, 11):
```

```
    # Get a random number between 1 and  
    the value of speed.
```

```
    vel1 = random.randint(1, speed)
```

Continued on next page...

Loops (For) pt 5

...Continued from previous page.

```
# Whatever distance is right now, add  
vel1 to it.
```

```
distance1 = distance1 + vel1
```

```
# How much distance does the car  
have? That's how many spaces we want  
behind it.
```

```
print((space * distance1) + car)
```

```
vel2 = random.randint(1, speed)
```

```
distance2 = distance2 + vel2
```

```
print((space * distance2) + car)
```

```
# What "round" are we on?
```

```
print(x)
```

```
print(roadside)
```

```
# Wait 1/3 of a second before we keep  
going.
```

```
time.sleep(0.3)
```

Loops (For) pt 6



Explanation:

`import random` tells Python that we want to use the tools Python has for working with random numbers.

`import time` tells Python that we want to use the tools it has for working with time.

Then, we define some key variables at the top so they're easier to find and change later.

Our for loop will run 10 times. Each time, it does these things:

1. Makes a new variable called `vel1` which is a random number between 1 and the value of `speed`.

Loops (For) pt 7

2. Adds vel1 to whatever distance is right now. If on the first run vel1 is 3, then this part of the code will mean $\text{distance1} = 0 + 3$. If it is 2 on the next run, the code will mean $\text{distance1} = 3 + 2$. In this way, the distance1 variable gets bigger each time the loop runs.

3. Makes space behind the car. How much space? It depends on the distance that car has traveled. It's actually possible to multiply strings! If we want to print "yo" three times, we can do so like this:

```
print("yo" * 3)
```

Loops (For) pt 8

4. Repeats steps 1 - 3 for the second car.

5. Prints out the round number, which is stored in the variable x.

6. Prints some dashes that look like the side of a road to make the output easier to understand.

7. Tells Python to wait for about $\frac{1}{3}$ of a second before it continues running the program. This way we can see our cars moving!