



# Introduction to For Loops

# For loops

```
for i in range(0, 5, 1):  
    print(i)
```

Like while loops, used to repeat a code block

Use to repeat a fixed number of times

One way to define that fixed number is with the range() function

## For loops with range() function

```
for i in range(0, 5, 1):  
    print(i)
```

**Initial value of i**

```
for <iter var> in range(start, stop, step):  
    statement 1  
    statement 2  
    etc
```

## For loops with range() function

```
for i in range(0, 5, 1):  
    print(i)
```

**Initial value of i**

```
for <iter var> in range(start, stop, step):  
    statement 1  
    statement 2  
    etc
```

## For loops with range() function

```
for i in range(0, 5, 1):  
    print(i)
```

**max value of i**

```
for <iter var> in range(start, stop, step):  
    statement 1  
    statement 2  
    etc
```

## For loops with range() function

```
for i in range(0, 5, 1):  
    print(i)
```

**increment/decrement**

```
for <iter var> in range(start, stop, step):  
    statement 1  
    statement 2  
    etc
```

 for-range-loop.py ×

1-Fundamentals >  for-range-loop.py > ...

```
1 for i in range(0, 5, 1):
2     print(i)
3
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

1: bash

ariadne:1-Fundamentals minae\$ python for-range-loop.py

```
0
1
2
3
4
```

ariadne:1-Fundamentals minae\$

# For loops with iterable value



Similar to JavaScript for ... of loops

Provide for loop with an iterable value, e.g. list, tuple, dictionary, set, string

For loop iterates once for each item in a list, key in a dictionary, character in a string, etc, then stops at the end



 for-range-loop.py ×

1-Fundamentals >  for-range-loop.py > ...

```
1  for x in "string":
2      print(x)
3
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

1: bash

ariadne:1-Fundamentals minae\$ python for-range-loop.py

s  
t  
r  
i  
n  
g

ariadne:1-Fundamentals minae\$