

Section Overview

What You Will Learn

- Numbers
- Numeric Operations
- Integers
- Floats
- Comments

Numbers

Numbers

- Use numbers directly in your source code
 - Do not use quotation marks as they are for strings.

```
integer = 42
```

```
float = 4.2
```

Numeric Operations

Symbol	Operation
+	add
-	subtract
*	multiply
/	divide
**	exponentiate
%	modulo

Exponentiation

$2^{**}4$ means "2 raised to the power of 4"

$$2 * 2 * 2 * 2$$

Modulo Operator

$$3 \% 2 = 1$$

$$4 \% 2 = 0$$



Python 3.4.3 (default, Oct 14 2015, 20:28:29)

[GCC 4.8.4] on linux

Type "copyright", "credits" or "license()" for more information.

>>> 1 + 2

3

>>> 3 % 2

1

>>> 8 / 4

2.0

>>> |


```
sum = 1 + 2
```

```
difference = 100 - 1
```

```
product = 3 * 4
```

```
quotient = 8 / 2
```

```
power = 2 ** 4
```

```
remainder = 3 % 2
```

```
print('Sum: {}'.format(sum))
```

```
print('Difference: {}'.format(difference))
```

```
print('Product: {}'.format(product))
```

```
print('Quotient: {}'.format(quotient))
```

```
print('Power: {}'.format(power))
```

```
print('Remainder: {}'.format(remainder))
```

Sum: 3

Difference: 99

Product: 12

Quotient: 4.0

Power: 16

Remainder: 1

Floating Point Numbers (Floats)

$$8 / 2 = 4.0$$

$$1 + 2.0 = 3.0$$

```
sum = 1 + 2  
difference = 100 - 1  
new_number = sum + difference  
print(new_number)  
print(sum / sum)  
print(sum + 1)
```

102

1.0

4

Strings and Numbers

```
quantity = 3
quantity_string = '3'
total = quantity_string + 2
```

```
Traceback (most recent call last):
```

```
  File "string_test.py", line 3, in <module>
```

```
    total = quantity_string + 2
```

```
TypeError: Can't convert 'int' object to str implicitly
```

The int() function

```
quantity_string = '3'  
total = int(quantity_string) + 2  
print(total)
```

5

The float() function

```
quantity_string = '3'  
quantity_float = float(quantity_string)  
print(quantity_float)
```

3.0

Comments

Comments

```
# This is a comment. Python ignores them.
```

```
# The following code:
```

```
#     Computes hosting costs.
```

```
#     Determines the duration of hosting
```

```
#         that can be purchased given a
```

```
#         budget.
```

Comments

```
""" This is the start of the comment  
This is another line.  
This is the last line in the comment. """
```

```
"""
```

```
I've started this comment down here.  
Python will not try to interpret these lines  
since they are comments.
```

```
"""
```

Comments

```
"""This is yet another comment."""
```

Comments

```
# Get the input from the user.
text = input('What would you like the cat to say? ')

# Determine the length of the input.
text_length = len(text)

# Make the border the same size as the input.
print('          {}'.format('_' * text_length))
print('        < {} >'.format(text))
print('          {}'.format('-' * text_length))
```

Section Summary

Summary

- Unlike strings, numbers require no special decoration.
- If you enclose a number in quotes it is actually a string.

Summary

- To convert a string to an integer, use the `int()` function.
- To convert a string to a float, use the `float()` function.

Summary

- Single line comments begin with an octothorpe (#).
- Multi-line comments are enclosed in triple quotes (""").