

# Python Programming

John Dunnion

School of Computer Science  
University College Dublin

## Numbers



## Numbers in Python programs



## Python versions

# Outline

## Numbers

### Expressions

### Arithmetic operators in Python

### Division in Python

### Powers

### Variables and assignment

### The variable `_`

## Numbers in Python programs

### Using numbers in Python programs

### Importing the `math` module

### Importing modules

## Python versions



## Expressions

- The Python interpreter can act as a simple calculator
- When you type an expression (eg  $4 + 6$ ), the interpreter **evaluates** the expression and prints out the value (eg 10)
- The operators `+`, `-`, `*` and `/` work just like in most programming languages, eg C and Java
- Parentheses `((` and `))` can be used to group **sub-expressions**
- Expressions in Python have a particular **type**
- Whole numbers (integers) are represented in Python using the type **int**
- Numbers with a fractional part (real numbers) are represented in Python using the type **float**



## Arithmetic operators in Python

Python Operator	Operation
+	Addition
-	Subtraction
*	Multiplication
/	[Floating-point] Division
//	Integer Division
%	Remainder after integer division
**	Power



## Python Expressions (1)

```
>>> 2 + 2  
4  
>>> 50 * 4  
200  
>>> 4 * 3 + 2  
14  
>>> 4 * (3 + 2)  
20
```



## Python Expressions (2)

- The integer numbers (eg 1, 2, 20, 20000000) have type **int**
- Numbers with a fractional part (eg 1.5, 2.444, 20.0) have type **float**
- Expressions with mixed type operands convert the integer operand to floating-point



## Python Expressions (3)

```
>>> 1 + 1  
2  
>>> 4 + 4  
8  
>>> 20.5 + 42.1234  
62.6234  
>>> 1234.5 + 765.5  
2000.0  
>>> 50 * 5  
250  
>>> 50 * 5.0  
250.0  
>>> 234.5 * 15  
3517.5
```



## Division in Python

- Division (/) in Python 3.x **always returns a float**



## Division in Python

```
[john@localhost ~]$ python3
```

```
>>> 6 / 3
```

```
2.0
```

```
>>> 7 / 3
```

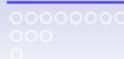
```
2.333333333333335
```

```
>>> 6 / 3.0
```

```
2.0
```

## Division and “Integer Division” (1)

- Division (`/`) in Python 3.x **always** returns a float
- To do integer division (“floor division”) and always get an `int` result, use the `//` operator
- To get the remainder after integer division, use the `%` operator

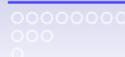


## Division and “Integer Division” (2)

```
[john@ localhost ~] $ python3
```

```
>>> 23 / 3  
7.666666666666667
```

```
>>> 23 // 3  
7  
>>> 23 % 3  
2  
>>> 7 * 3 + 2      # result * divisor + remainder  
23
```



## Powers

- The “`**`” operator can be used to calculate powers

```
>>> 3 ** 2      # 3 squared  
9  
>>> 2 ** 8      # 2 to the power of 8  
256
```