Topic 1.2: Python
Numbers and Strings
Aims

• Understand values that can be used in Python
  • Numbers
  • Strings
• Write expressions using numbers and strings
• Understand why some expressions give errors

Try out the example in this topic using the Python shell
Values – Numbers

• Python can work with numbers

• Integers: 10  20  12345
• Decimals: 7.2  -0.00134
Values – Strings

• A string is a sequence of letter
  • ’David Cameron’
  • ”David Cameron”

• Either single or double quotes can be used but must be the same at either end
  • This allow quote characters in strings
Arithmetic

- Python does arithmetic using operators:

<table>
<thead>
<tr>
<th>Operation</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>x + y</td>
<td>sum of x and y</td>
</tr>
<tr>
<td>x - y</td>
<td>difference of x and y</td>
</tr>
<tr>
<td>x * y</td>
<td>product of x and y</td>
</tr>
<tr>
<td>x / y</td>
<td>x divided by y</td>
</tr>
<tr>
<td>x // y</td>
<td>integer division of x by y</td>
</tr>
<tr>
<td>x % y</td>
<td>remainder of x / y</td>
</tr>
<tr>
<td>-x</td>
<td>x negated</td>
</tr>
<tr>
<td>+x</td>
<td>x unchanged</td>
</tr>
<tr>
<td>pow(x, y)</td>
<td>x to the power y</td>
</tr>
<tr>
<td>x ** y</td>
<td>x to the power y</td>
</tr>
</tbody>
</table>
Arithmetic

- Brackets can be used:
  - $10 - 5 - 2 = 3$ left to right order
  - $10 - (5 - 2) = 7$ … as in maths
- Two kinds of division
  - $10 / 4 = 2.5$ real (floating point) division
  - $10 // 4 = 2$ integer division
- The ‘%’ operator mean remainder
  - $10 \% 3 = 1$
String Concatenation

- Python can join strings together

'Hello' + 'World' gives 'HelloWorld'

'Hello' + " " + "World" gives 'Hello World'

- Notice that the same operator '+' has two different uses
  - Adding numbers
  - Joining string
Indexing and Slicing Strings

- Indexing get a character from a string
  - "william"[0] – gives ‘W’
  - "william"[1] – gives ‘i’
  - "william"[6] – gives ‘m’
- Notice that numbering starts from zero

- Slicing is used to get a subrange
  - "william"[1:4] – gives ‘ill’
- Notice that the slice [N:M] includes N but not M
String Length

• You can find the length of a string using ‘len(’
  • `len("William")` – gives 7
  • `len("")` – gives 0
Finding a Character

- Where is a character in a string? Use the `.index()` method
  - “David Cameron”.index(“ “) – give 5
  - “David Cameron”.index(“a”) – give 1

- Note: index finds the first character

Technical Note
Why is ‘.index()’ written in a different way to ‘len()’?
  - ‘len’ is a built in function
  - ‘.index()’ is a method
This difference will become clearer later.
Errors

• Python has fewer errors than other languages (e.g. Java)
  • This has both pros and cons
• Not everything we write makes sense
• Syntax error: “I can’t understand what you are asking”
  • 123abc – not a number
  • 1 ! 3 – not an operator
  • “hello – a string with no end
Evaluation Errors

• “The text looks ok but when I try to calculate, it makes no sense”
  • 42 + “hello” – can’t combine a number and a string
  • 42 / 0 – can’t divide by zero
  • “hello”[17] – can’t index beyond the end
Types

- All values belong to a particular type
- Strings
- Numbers
  - Integer
  - Floating point
- An operator works for values of the correct type or types
Input a Number
Input a Number

- Input always reads a string
  - Must not confuse string and number
- Consider:

```python
# This program calculates your age next year
# ... unfortunately it does not work
age = input("How old are you? ")
print("Next year you will be", age+1)
```

- The result is:

```
How old are you? 21
Traceback (most recent call last):
  File "age-wrong.py", line 4, in <module>
    print("Next year you will be", age+1)
TypeError: Can't convert 'int' object to str implicitly
```
Using the ‘int’ function

- Use the ‘int’ function to convert a string (of digits) to a number
- Try the corrected program:

```python
#This program calculates your age next year
age = input("How old are you? ")
print("Next year you will be", int(age)+1)
```
Where do Functions Come From?

Python Library
Functions So Far

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>print</td>
<td>Write some output</td>
</tr>
<tr>
<td>input</td>
<td>Get keyboard input</td>
</tr>
<tr>
<td>len</td>
<td>Find length of a string</td>
</tr>
<tr>
<td>int</td>
<td>Convert string to integer</td>
</tr>
</tbody>
</table>

• Where do the functions come from?
• How do we find about these functions?
Library

- Software written (and tested) by someone else for you to use in your program
- Good news
- BUT
  - More complex than the language
  - Just learn the bits you need
Python v3.2.3 documentation

Welcome! This is the documentation for Python 3.2.3, last updated Oct 28, 2012.

Parts of the documentation:

- What's new in Python 3.2?
  or all "What's new" documents since 2.0

- Extending and Embedding
  tutorial for C/C++ programmers

- Tutorial
  start here

- Python/C API
  reference for C/C++ programmers

- Library Reference
  keep this under your pillow

- Installing Python Modules
  information for installers & sysadmins

- Language Reference
  describes syntax and language elements

- Distributing Python Modules
  sharing modules with others

- Python Setup and Usage
  how to use Python on different platforms

- FAQs
  frequently asked questions (with answers!)

- Python HOWTOs
  in-depth documents on specific topics
Built-In Functions

• Look at the documentation for built in functions: http://docs.python.org/3.2/library/functions.html
  • It may be available on your machine, from the IDE
• Look up the functions we have already covered
  • print, input, len, int
• Also look at:
  • min, max, ord, chr, pow, abs, bin
• … and string methods
  • http://docs.python.org/3.2/library/stdtypes.html#string-methods
Summary

• Introduced some Python expressions
• An expression has a value
• … unless it creates an error

• Some expressions are numbers
• … others and string
• … more to learn about.

• There is a library
  • Good news: lots of code for you to use