Methods

Wrestling with Python

Overview

- The Story so far….
- Creating methods to make programs simpler
- Returning a value from a method
- Sending values into a method

THE STORY SO FAR

What we can do so far

- Store data (using variables)
- Change data (using assignments)
- Make decisions (using conditions)
- Loop round statements(using while)
- Loop through sequences(using for)
- Store collections of data(using lists)

METHODS

Methods in Pyton

- We have been using methods in Python for some time

  `input` - reads something from the user

  `print` - prints a message

  `int` - converts a string into an integer

- These are all methods which we have been calling to do tasks for us

What is a method?

- A method is a block of code that you can refer to by its identifier
- Whenever you refer to the method the block of code that is in the method is executed for you
- Methods let you break a solution down into smaller parts

Our first method

```
def silly():
     print("hello from silly")
```

- This method is called `silly`
- The body just prints a message

Using a method

```
silly()
print("and we are back")
```

- You call a method just by giving its name
- The program runs the method and then returns to the program so that the message is printed

What happened here?

```
bicycle()
Traceback (most recent call last):
  File "<pyshell#12>", line 1, in <module>
    bicycle()
NameError: name 'bicycle' is not defined
```

What happened here?

```
bicycle()
Traceback (most recent call last):
  File "<pyshell#12>", line 1, in <module>
    bicycle()
NameError: name 'bicycle' is not defined
```

- The method “bicycle” does not exist, and so python was unable to find and execute it
- You must define a method before you use it

RETURNING VALUES

Returning a value

- It is often useful to make a method that returns a value
- We could create one that returns a number that the user has typed in
- Python allows a method to return a value
- It uses the return keyword to do this

Returning a result

```
def FetchANumber():
        numberString = input()
        return int(numberString)
```

- The method called `FetchANumber` will fetch a number from the user and return it
- It gets the number as a string and then converts it into an integer

PARAMETERS

Parameters

- Fetching a number is all very well, but we really need a way that the method can display a prompt
- Methods can be given one or more *parameters* which are copied into the method when the method is called

Method with parameter

```
def FetchANumber(promptString):
     numberString = input(promptString)
     return int(numberString)
```

- The name of the parameter is supplied in brackets when the method is declared
- We can use the parameter value inside the method code

Using the parameter

```
def FetchANumber(promptString):
    numberString = input(promptString)
    return int(numberString)
```

- Everywhere the parameter is used the value given when the method was called is used

Default Parameters

```
def FetchANumber(promptString="Enter a number:"):
     numberString = input(promptString)
     return int(numberString)
```

- You can give a value for a parameter that will be substituted if the parameter is not supplied when the method is called
- If we don’t give a prompt string the method above will use “Enter a number:”

Calling a method

```
age = FetchANumber("Enter your age: ")
```

- When the method is called the value of the parameter is passed into it
- This would cause the method to ask the user for an age value
- We have been using Python methods that accept parameters for some time

Multiple Parameters

```
def sum(x,y):
    return x + y
```

- You can create a method with more than one parameter if you wish
- The above method accepts two items and adds them together

Methods and Madness

```
result = sum(2,3)
```

- This would set the value of result to 5, because that is what you get when you add the value 2 to the value 3

Methods and Madness

```
result = sum("hello ","world")
```

- This would set the value of result to “hello world” because that is what you get when you add the value “hello ” to the value “world”

Methods and Madness

```
result = sum("hello ",99)
```

- What would this do?

Methods and Madness

```
result = sum(2, "hello ")
```

- What would this do?
- It would fail, since Python does not know how to add a string to an integer:

```
TypeError: unsupported operand
type(s) for +: 'int' and 'str'
```

Golden Rule

- If a method has multiple parameters you must make sure that your call to the method “makes sense”
- Otherwise the program will fail when it runs
  - Or do stupid things

Making A Useful Method

```
NoOfPlayers = ReadNumber("Enter number of players : ", 2, 4)
```

- This method can be used to read a value
- It is given a prompt string and the min and max values of the range
- It then returns with a result in that range
- You are going to write this

Summary

- Methods allow us to break a program into a smaller chunks and reuse code
- A method is defined to have a particular name which is used when it is called
- A method can receive values to work on (parameters) and return a result