You're flying! How?

Python!

I learned it last night! Everything is so simple!
Hello world is just:
print "Hello, world!"

I dunno... dynamic typing? Whitespace?
Come join us! Programming is fun again!
It's a whole new world up here!
But how are you flying?

I just typed import antigravity
That's it?

... I also sampled everything in the medicine cabinet for comparison.
But I think this is the Python.

CSCI 141
Scott Wehrwein
Importing Modules
The random module
Goals

• Know how to import a module and call its functions.

• Know how to generate random numbers using the random module's randint function.
Other Peoples’ Code

We’ve already used code other people wrote by calling built-in Python functions:

- print, input, type

Built-in functions are special because they’re always available.

Many other functions exist in the Python Standard Library, which is a collection of modules containing many more functions.
Other Peoples’ Code

An example: I want to generate a random integer between 0 and 10.

I don’t know how to do this.
Someone who does has written some functions for me. They live in the random module:

    import random

I could go look at the source code...
An example: I want to generate a random integer between 0 and 10.

```python
import random

# This code is a bit messy to make it fast for the common case while still doing adequate error checking.
_istart = _int(start)
if _istart != start:
    raise ValueError("non-integer arg 1 for randrange()")
if stop is None:
    if _istart > 0:
        return _istart + self._randbelow(_istart)
    raise ValueError("empty range for randrange()")

# stop argument supplied.
_istop = _int(stop)
if _istop != stop:
    raise ValueError("non-integer stop for randrange()")
width = _istop - _istart
if step == 1 and width > 0:
    return _istart + self._randbelow(width)
if step == 1:
    raise ValueError("empty range for randrange() (\%d, \%d, \%d)" % (_istart, _istop, width))

# Non-unit step argument supplied.
_istep = _int(step)
if _istep != step:
    raise ValueError("non-integer step for randrange()")
```

This fixes the problem with randint() which includes the endpoint; in Python this is usually not what you want.

------
An example: I want to generate a random integer between 0 and 10.

I don’t know how to do this.
Someone who does has written some functions for me.
They live in the `random` module:

```python
import random

num = random.randint(0, 10)
```

I could go look at the source code... but I’d rather just use their functions without knowing how they work.
Other Peoples’ Code

```python
import random
num = random.randint(0,10)
```

Two questions:

1. **What is this syntax about?**

2. **How do I know what the function does?**
Using Modules: Syntax

The Python Standard Library is a collection of modules containing many more functions.

To use functions in a module, you need to import the module using an import statement:

```
import module
```

(replace the text in this font with the specific module name)

By convention, we put all import statements at the top of programs.
Using Modules: Syntax

Once you’ve imported a module:

```python
import random
```

you can call functions in that module using the following syntax:

```python
random.randint(0, 10)
```

Module name  Dot  Function call (the usual syntax)
Two questions:

1. What is this syntax about?

2. How do I know what the function does?
Other Peoples’ Code

```
import random

num = random.randint(0,10)
```

Two questions:

1. What is this syntax about?

2. How do I know what the function does?

Read about it in the Python documentation.

My approach, in practice:

1. Google “python 3 <whatever>”
2. Make sure the URL is from python.org and has version python 3.x
Demo
Demo

- guess.py
  - Pick a random number between 1 and 10
  - Count how many tries it takes a user to guess it.
math module

• The math module has useful stuff!

• You can read about it in the documentation.

• logarithms, trigonometry, ...

• Modules can also contain values:

```python
>>> import math
>>> math.pi
3.141592653589793
>>> math.e
2.718281828459045
```
More on import statements

• Import the entire module:

```python
import random
num = random.randint(1, 10)
```

• Import a specific function:

```python
from math import sin
sin0 = sin(0)
```

• Don’t need module name dot notation
• Other `math` functions are not accessible
Demo

- Thonny shell:
  - import math
  - from math import

- wave.py
  - draw a sine-wavy picture using text art