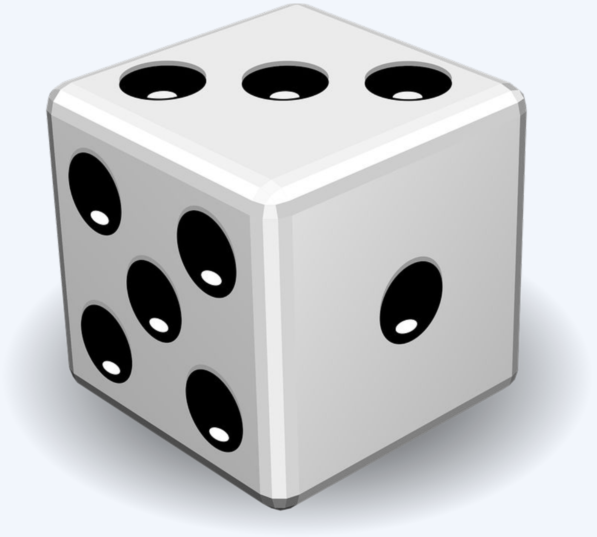
Lesson 4: Randomisation

Roll the dice!

Question

Why do you use a dice in real-life games?

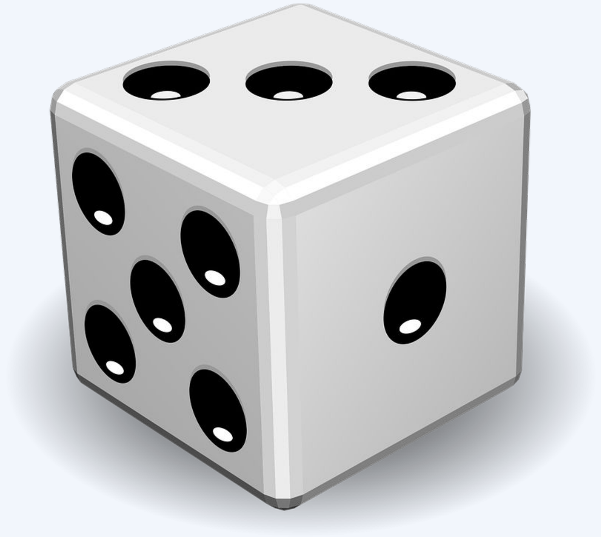


Roll the dice!

When you roll a dice by hand, the result is determined by many things such as:

- Starting position
- Friction
- Gravity
- Air resistance

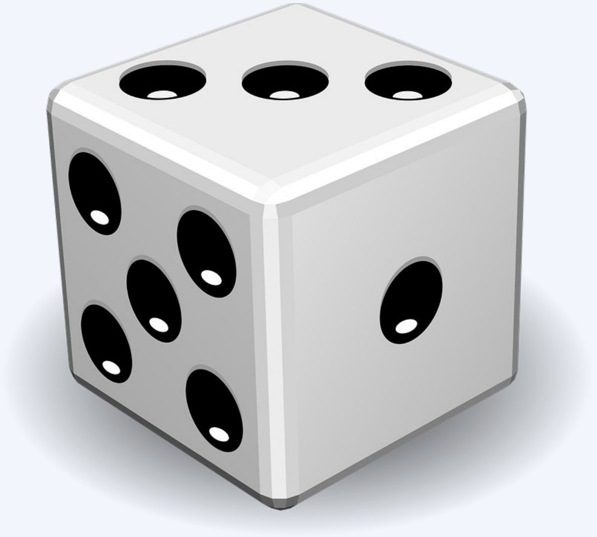
The outcome of the resulting throw is therefore pretty **random**.



Roll the dice!

Question

Where might you need to generate **randomness** in a computer game?



Roll the dice!

Any dice game

To generate random words for a word guessing game

To play a game like rock, paper, scissors against a computer

Lesson 6: Randomisation

In this lesson, you will:

- Be able to locate information using the Python documentation
- Import modules into your Python code
- Demonstrate how to generate random numbers in Python

Make a prediction (think, pair, share)

```
from random import randint

number = randint(1,5)
print(number)
```

Question

What will be the output of `print`, when this program is executed?

A. `number`
B. `randint(1,5)`
C. It is not possible to know the output without executing the program
D. `3`

Make a prediction (think, pair, share)

```python
from random import randint

number = randint(1,5)
print(number)
```

Question

What will be the output of `print`, when this program is executed?

A. `number`
B. `randint(1,5)`
C. It is not possible to know the output without executing the program
D. 3

Note: This is because a random number is being generated.

Importing the `randint` function

```python
from random import randint

number = randint(1,5)
print(number)
```

Let’s take a look at this code.

The first line is importing the function, **`randint`** from the module, **`random`**.

Importing the `randint` function

```
from random import randint

number = randint(1,5)
print(number)
```

Next, a function call has been assigned to **`number`**.

This **function** will take the **parameters** 1 and **5** and return a random integer from 1 to 5.

Importing the `randint` function

```
from random import randint

number = randint(1,5)
print(number)
```

The random integer will then be assigned to the variable **`number`**.

Importing the `randint` function

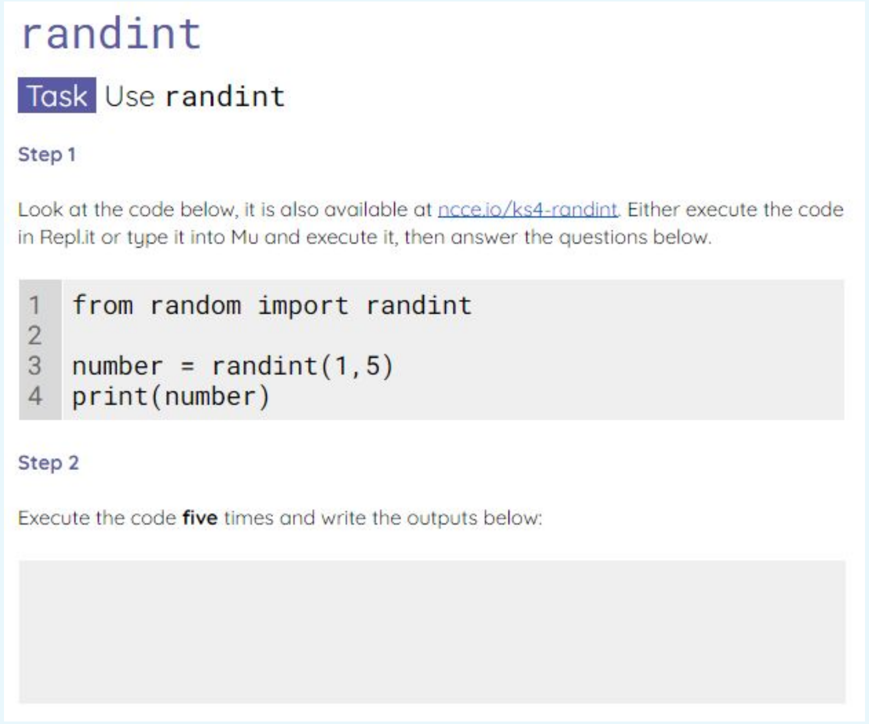
```python
from random import randint

number = randint(1,5)
print(number)
```

The random integer is then displayed as output for the user.

Try it yourself!

Use the **activity 1 worksheet** to run and investigate this piece of code.



randint

Task Use `randint`

Step 1

Look at the code below, it is also available at ncce.io/ks4-randint. Either execute the code in Repl.it or type it into Mu and execute it, then answer the questions below.

```
from random import randint

number = randint(1,5)
print(number)
```

Step 2

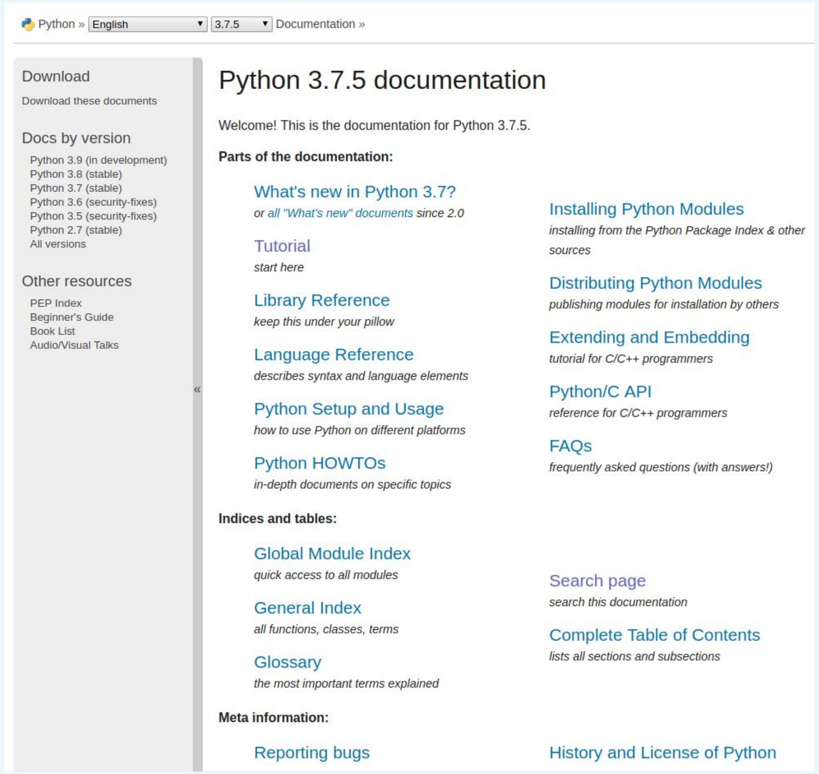
Execute the code **five** times and write the outputs below:

Python documentation

Programming languages come with documentation to help developers get the most out of the language.

Developers very rarely know all the functionality of a language.

Accessing this documentation is commonplace for finding out how to perform a particular function within a language.



Python » English 3.7.5 Documentation »

Download
Download these documents

Docs by version
Python 3.9 (in development)
Python 3.8 (stable)
Python 3.7 (stable)
Python 3.6 (security-fixes)
Python 3.5 (security-fixes)
Python 2.7 (stable)
All versions

Other resources
PEP Index
Beginner's Guide
Book List
Audio/Visual Talks

Python 3.7.5 documentation

Welcome! This is the documentation for Python 3.7.5.

Parts of the documentation:

What's new in Python 3.7?
or all "What's new" documents since 2.0

Tutorial
start here

Library Reference
keep this under your pillow

Language Reference
describes syntax and language elements

Python Setup and Usage
how to use Python on different platforms

Python HOWTOs
in-depth documents on specific topics

Installing Python Modules
installing from the Python Package Index & other sources

Distributing Python Modules
publishing modules for installation by others

Extending and Embedding
tutorial for C/C++ programmers

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

FAQs
frequently asked questions (with answers!)

Indices and tables:

Global Module Index
quick access to all modules

General Index
all functions, classes, terms

Glossary
the most important terms explained

Search page
search this documentation

Complete Table of Contents
lists all sections and subsections

Meta information:

Reporting bugs

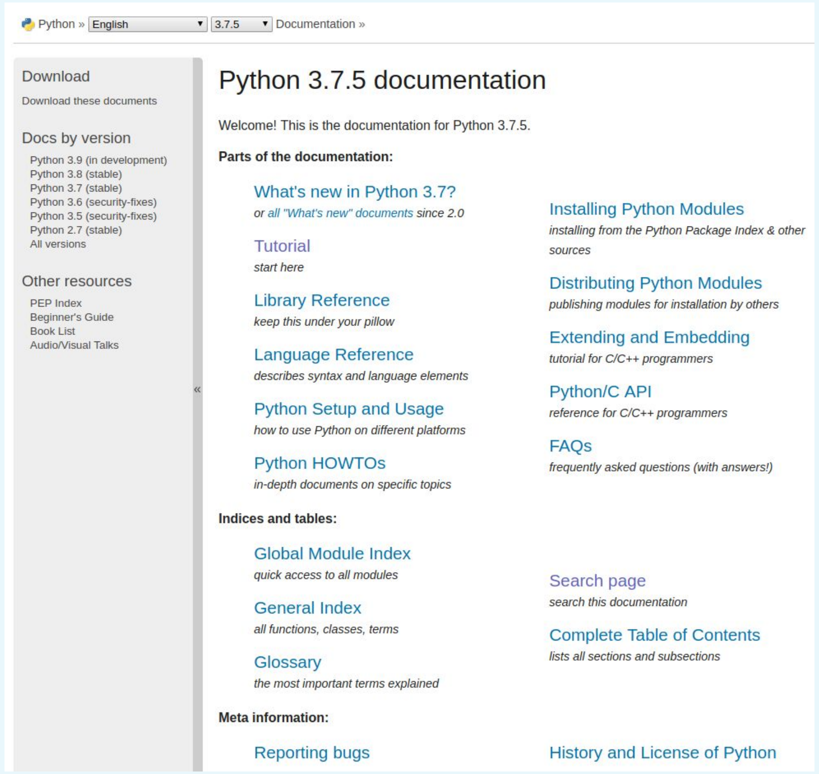
History and License of Python

Python documentation

The previous piece of code used the `random` module.

To find out specifically what this module does (plus lots of others and their relating functionality), you can visit the following link:

https://docs.python.org/3.7/



Python » English 3.7.5 Documentation »

Download

Download these documents

Docs by version

Python 3.9 (in development)
Python 3.8 (stable)
Python 3.7 (stable)
Python 3.6 (security-fixes)
Python 3.5 (security-fixes)
Python 2.7 (stable)
All versions

Other resources

PEP Index
Beginner's Guide
Book List
Audio/Visual Talks

Python 3.7.5 documentation

Welcome! This is the documentation for Python 3.7.5.

Parts of the documentation:

What's new in Python 3.7?
or all "What's new" documents since 2.0

Tutorial
start here

Library Reference
keep this under your pillow

Language Reference
describes syntax and language elements

Python Setup and Usage
how to use Python on different platforms

Python HOWTOs
in-depth documents on specific topics

Installing Python Modules
installing from the Python Package Index & other sources

Distributing Python Modules
publishing modules for installation by others

Extending and Embedding
tutorial for C/C++ programmers

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

FAQs
frequently asked questions (with answers!)

Indices and tables:

Global Module Index
quick access to all modules

General Index
all functions, classes, terms

Glossary
the most important terms explained

Search page
search this documentation

Complete Table of Contents
lists all sections and subsections

Meta information:

Reporting bugs

History and License of Python

Truly random vs pseudorandom

True random number generators

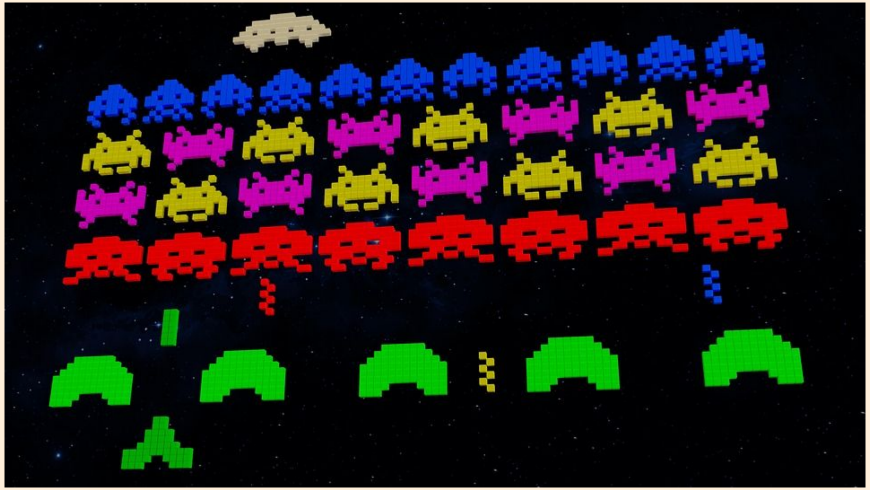
This uses unpredictable physical means to generate numbers, such as atmospheric noise. As these are truly random and are suitable for use in cryptography (keeping data secret!).

Truly random vs pseudorandom

Pseudorandom number generators

This uses mathematical algorithms which are computer-generated and therefore highly predictable.

Unless you are generating random numbers for security purposes, a pseudorandom generator would be fine, for instance in a computer game.



Spot the difference

Question

Identify the two differences between the two snippets of code below:

```python
import random

number = random.randint(1,5)
print(number)
```

```python
from random import choice

number = randint(1,5)
print(number)
```

The two lines highlighted indicate the different ways in which `import` can be used with modules/functions.

`import` module vs `from` module import

The code below imports the whole module and references the module AND the function when you want to use it.

```
import random

number = random.randint(1,5)
print(number)
```

The code below imports the specific function from a module and references them directly in the code.

```
from random import choice

number = randint(1,5)
print(number)
```

Using this method allows you to **only import what you need**.

Can you determine what is missing?

```
1   from random import randint
2   number = randint(0,10)
3   print(number)
```

Question

Look carefully at this program. What is missing from line 1 to make the code work?

[✓ 1] `from random import randint`
2 `import random`
3 `import randint`
4 `from random import choice`

On line 2 the code directly states **`randint`** rather than **`random.randint`** so you can tell you are only importing and using what you need.

Check out the comments

Commenting code is really important. It allows you to explain what your code does to someone who may read it if you are not there to explain it, and also to yourself in the future as an aid.

```
‘’’If you want a comment to
span multiple lines, then use
a triple quote at the start
and end like this.’’’

print(“Comments are useful!”)

#Comments on a single line
#start with a hash symbol
```

Comment your code

Add suitable comments to your code to explain what your previous program did.

Be sure to use the most appropriate type of comment, either multi-line ''' ''' or single line #.

Comments generally start on the line above the line of code you are talking about.

```python
from random import randint

my_number = randint(0, 10)

print(my_number)
```

Another function from random

This code imports a different function called **`randrange.`** Use the link below to access the code and complete the ‘Another function’ worksheet.

ncce.io/randrange

```
from random import randrange
my_number = randrange(0, 10, 2)
print(my_number)
```

The structure of `randrange`

The `randrange` function allows you to choose a **minimum** and **maximum** value, additionally the last parameter in the expression states the **step** value (what to count up in from the minimum).

```python
from random import randrange
my_number = randrange(0, 10, 2)
print(my_number)
```

The structure of randrange

In this example, the step value 2 ensures the number output will always be even.

This could be changed to ‘count up’ in any number within the range stated.

```python
from random import randrange
my_number = randrange(0, 10, 2)
print(my_number)
```

Changing the values

The **minimum**, **maximum**, and **step** values have been changed in this example.

```python
from random import randrange
number = randrange(20, 100, 7)
print(number)
```

Question

How will the changes in values affect the output, and what would be a potential output of the program?

Changing the values

The lowest output will be **20** and the highest output will be **100**.

Anything in between these numbers will be in increments of **7**.

Example

27, **41**, **62** would be examples of valid outputs from the code.

```python
from random import randrange
number = randrange(20, 100, 7)
print(number)
```

Pair the parameters!

Match the **randrange** statements to the outputs:

```
num=randrange(0, 50, 5)                             128

num=randrange(13, 66, 2)                            12

num=randrange(100, 150, 4)                          35

num=randrange(0, 200, 7)                            37

num=randrange(5, 15, 1)                             119
```

Pair the parameters!

Match the `randrange` statements to the outputs:

$$\begin{tikzcd}
\texttt{num=randrange(0, 50, 5)} \arrow[rdd] & \texttt{128} \\
\texttt{num=randrange(13, 66, 2)} \arrow[rdd] & \texttt{12} \\
\texttt{num=randrange(100, 150, 4)} \arrow[ruu] & \texttt{35} \\
\texttt{num=randrange(0, 200, 7)} \arrow[rd] & \texttt{37} \\
\texttt{num=randrange(5, 15, 1)} \arrow[ruuu] & \texttt{119}
\end{tikzcd}$$

Next lesson

In this lesson, you…

Learnt how to locate information using the Python documentation

Imported modules into your Python code

Demonstrated how to generate random numbers in Python

Next lesson, you will…

Write and use expressions that use mathematical operators

Cast variables by calling a function returning a value of a desired data type

Assign expressions to variables