

```
user_number = input("Enter number: ")
user_input = input()
```

print("You entered", user_number, "and then", user_input)

Enter number: 98 ??? #user enters stuff here You entered 98 and then ???

The user input is shown in a pink colour

CompSci 101 - Principles of Programming

Random numbers

Quite often, in our programs, we need to generate random numbers, e.g., for games and simulations.

The random module contains a function, **randrange()**, which can be used to generate a random number. In order to use this function we need to import the random module into our program (just as we did when we wanted to use the functions defined in the math module – math.sin(), math.cos(), ...).

Whenever we need to get a random number in a program, the first line of the program will be the following import statement:

import random

Exercise

Complete the output if the user enters the number 54 when prompted:

user_number = input("Enter number: ")

print(user_number * 2, user_number * 3, user_number * 4)

Enter number: 54

CompSci 101 - Principles of Programming 8

Random numbers

The, **randrange()** function requires two values defining the range of the random number to be generated:

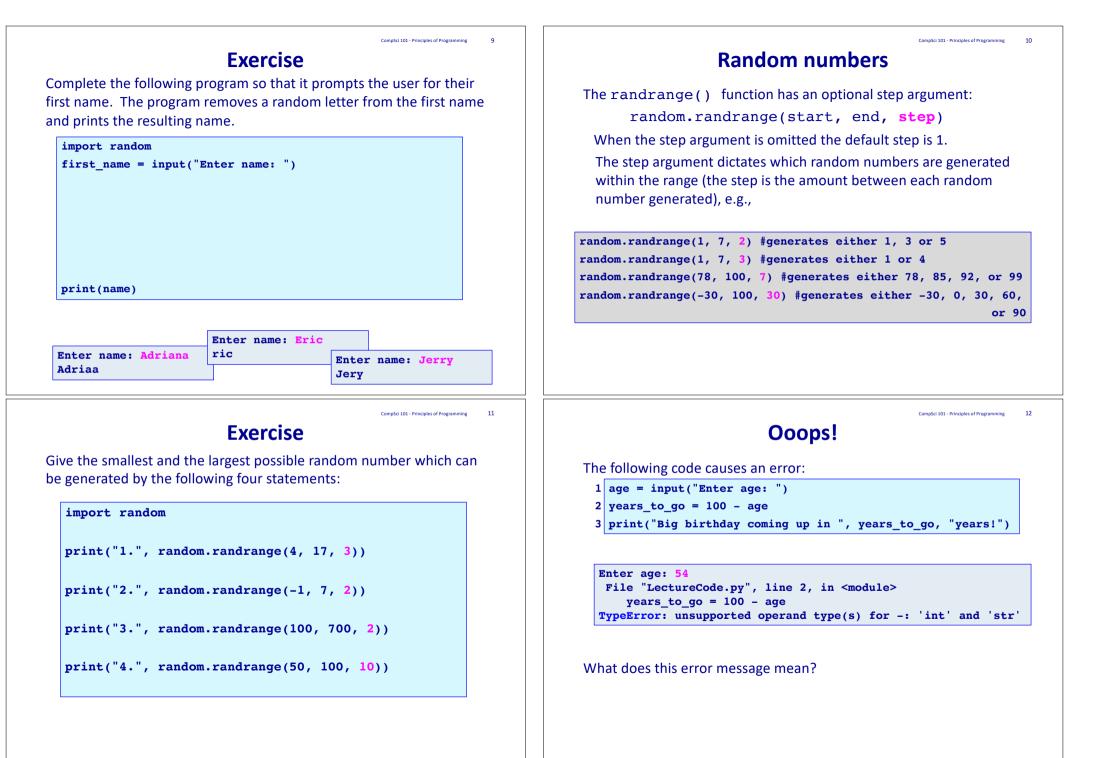
the first number is the lowest number which can be generated and the second number is one past the biggest number to be generated, e.g., random. randrange(5, 10) has an equal chance of generating 5, 6, 7, 8 or 9.

import random

```
dice1 = random.randrange(1, 7)
dice2 = random.randrange(1, 7)
```

print("You threw", dice1, "and a", dice2)

You threw 4 and a 1



Converting between types

The subtraction operator (-) has no meaning if one of the operands is a string. We want to find a way of converting a string containing just digits into a number.

The **int()** function converts a string containing characters which are digits into the corresponding integer value.

- 1 age = input("Enter age: ")
- 2 age = int(age)
- 3 years_to_go = 100 age
- 4 print("Big birthday coming up in", years_to_go, "years!")

Enter age: 54 Big birthday coming up in 46 years!

Note that the code on lines 1 and 2 can be combined into one line:

age = int(input("Enter age: "))

Converting between types

The conversion has to be legal. Below are two illegal attempts to convert values using int():

1 number = int("12 34")

File "LectureCode.py", line 1, in <module>
 number = int("12 34")
ValueError: invalid literal for int() with base
10: '12 34'

number = int("12.34")

```
File "LectureCode.py", line 1, in <module>
    number = int("12.34")
ValueError: invalid literal for int() with base
10: '12.34'
```

Note that converting a string which contains no decimal point into a float is fine:

```
number = float("12")
```

```
print(number)
```

12.0

str(comp dice)

Converting between types

Other functions which can be used to convert between types:

- float()str()
- 1 cost = input("Enter cost \$")
- 2 cost = float(cost)
- 3 final_price = cost * 0.92
- 4 print("Final price \$", final_price, sep="")

Enter cost \$509.59 Final price \$468.8228

String concatenation requires that the two operands be strings:

- 1 user_dice = input("Enter dice throw: ")
- 2 comp_dice = random.randrange(1, 7)

```
3 message = "Your dice: " + user_dice + ", computer dice: " +
```

```
4 print(message)
```

Enter dice throw: 5 Your dice: 5, computer dice: 1

Truncating a Floating Point Number

A floating point number can be truncated into the whole number part using int():

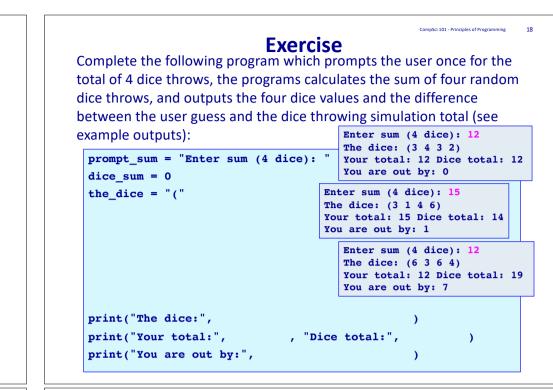
<pre>print("1:", int(12.34))</pre>	1: 12
<pre>print("2:", int(12.799))</pre>	2: 12
<pre>print("3:", int(12.999))</pre>	3: 12
<pre>print("4:", int(2.5))</pre>	4: 2
<pre>print("5:", int(3.5))</pre>	5: 3
<pre>print("6:", int(-6.213))</pre>	6: -6
<pre>print("0:", int(-0.213)) print("7:", int(-12.94))</pre>	7: -12
$p_{1110}(7.7, 100(-12.54))$	

import math

print("Truncating is not the same as flooring the number")
print("8:", math.floor(-6.213))

print("9:", math.floor(-12.94))

Truncating is not the same as flooring the number 8: -7 9: -13



Examples of Python features used in this lecture

```
dice1 = random.randrange(1, 7)
age = random.randrange(66, 99)
even_number = random.randrange(50, 99, 2)
tens = random.randrange(50, 101, 10)
```

```
user_input = input("Enter age: ")
age = int(user_input)
```

```
cost = input("Enter cost $")
cost = float(cost)
```

```
price = 32.45
message = "Final price $" + str(price)
```

Line continuation

The preferred way of line continuation (breaking long lines of code over multiple lines) is to use parentheses, brackets and braces.

	1	<pre>user_dice = input("Enter dice throw: ")</pre>
	2	<pre>comp_dice = random.randrange(1, 7)</pre>
There will be other examples of this in later lectures.	3	<pre>message = ("Your dice: " + user_dice +</pre>
4]		<pre>print(message)</pre>

The backslash character at the end of a line of code in the program also can be used to indicate that the statement continues onto the next line,

e.g.,

1 user_dice = input("Enter dice throw: ")

```
2 comp_dice = random.randrange(1, 7)
```

```
3 message = "Your dice: " + user_dice + ", computer dice: " + \
```

```
str(comp_dice)
```

4 print(message)

Note that the backslash is the last character on the line of code.

CompSci 101 - Principles of Programming

19

CompSci 101 - Principles of Programming

Summary

In a Python program:

- the input() function is used to get user input from the keyboard
- a random number can be generated using random.randrange(...)
- we can convert between types using str(), int(), float()