

# Project 2 - Random numbers

Write a program which:

- creates a random number between 1 and 20
- allows the user to guess a number
- tells the user if they are correct or not

Here's how to create a die throw random number:

```
import random #at the top of the code
```

```
#create random integer between 1 and 6
```

```
dieThrow = random.randint(1, 6)
```

# Project 2 - Random numbers

Here's how to create a die throw random number:

## Python:

```
import random #at the top of the code  
#create random integer between 1 and 6  
dieThrow = random.randint(1, 6)
```

## Pseudo code:

```
dieThrow <- RANDOM_INT(1, 6)
```

# Project 2 - Random numbers

Write a program which:

- displays the title “Random Guess Program”
- displays a message telling the user what the rules of the game are
- creates a random number between 1 and 20
- allows the user to guess a number
- tells the user if they are correct or whether their guess is higher or lower than the random number

# Project 2 - Random numbers

Write a program which:

- displays the title “Random Guess Program”
- displays a message telling the user what the rules of the game are
- creates a random number between 1 and 20
- allows the user to guess a number until they get the answer correct (hint: while loop)
  - after each guess, tell the user if they are correct or whether their guess is higher or lower than the random number
- tells the user how many guesses it took them

# Project 2 - Random numbers

Extension tasks:

- A. allow the user to specify the high value (rather than 20)
- B. check each time that the user has entered a valid number. If they haven't the program should ask for them to reenter it (hint: while loop and try-except)
- C. let the computer play the game instead of the user (see next slide)

# Project 2 - Random numbers

The computer will always use a **Binary Search** routine when it guesses.

This involves choosing the middle number. So, if the values are from 1 to 20, it will always guess 10 first. Then, depending on whether the response is higher or lower, it will go to 5 or 15.

There are issues though:

- numbers will need rounding - use `round(theNumber)`
- as soon as you divide a number, Python uses float data types, which may (or may not) cause an issue

Using Binary Search guarantees that you can always guess a number between 1 and 20 in no more than 5 guesses. For a value between 1 and 100 it's 7 guesses. For 1 to 1000 it's no more than 10 guesses...

# Project 2 - Random numbers

The other search method you need to know is a linear search

This simply starts from 1 and guesses in order - so the maximum number of guesses that could be needed is 20 (the minimum is 1)