Password Generator

Introduction:

It's important to protect your personal information online, and in this project you'll create a program to generate passwords for you.

The passwords will be random, so no one will be able to guess them!

```
Password Generator
==================

number of passwords? 4
password length? 15

here are your passwords:
V^?SyNd6?Sf^3ov
txWm(EM7g@1rrE0
Eyni2p%1(M2Bgi5
uI$XLC0mFWmB*GK
```

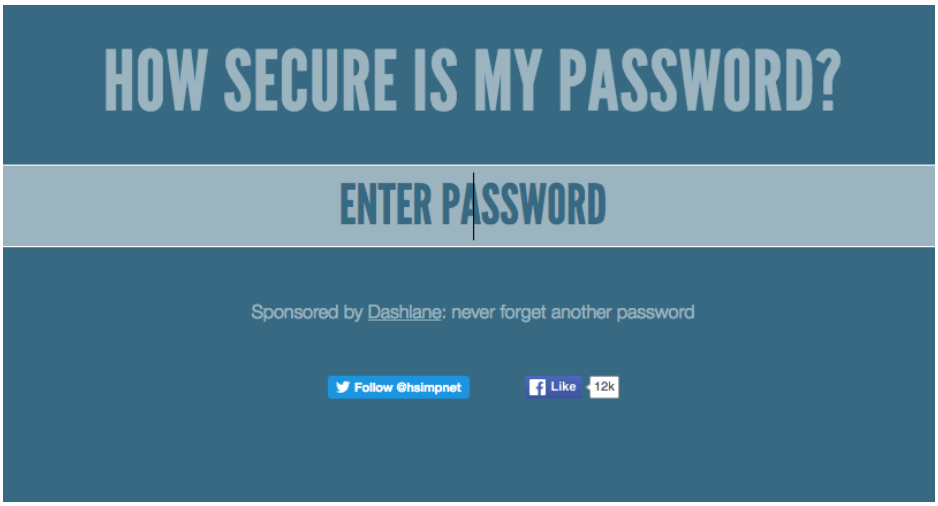
Step 1: How secure is your password?

A computer could try to guess your password by using 'brute force' – this means trying out lots of passwords until it guesses the right one.

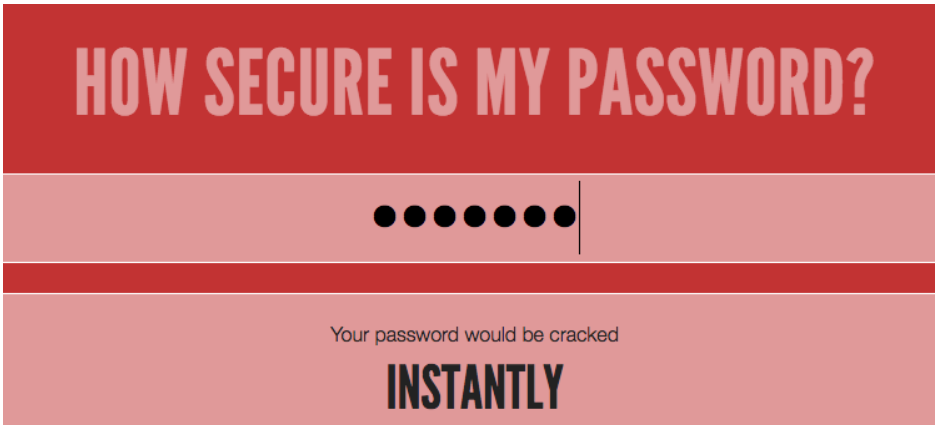
Let's find out how long it would take a computer to guess your password.

Activity Checklist

- [ ] Go to **howsecureismypassword.net**, which is a website for finding out how secure your passwords are.



- [ ] Type in “letmein” (Let me in) as the password. You’ll see that a computer would guess this password **instantly**!



You’ll also see some reasons why “letmein” isn’t a good password to use:

- It’s a very **common** password (one of the 15 most used passwords). A computer would guess these first.
- It contains words from the **dictionary**. A computer would also try these passwords first.
- It’s very **short**. It would take a computer more time to guess a longer password.
- It only contains **letters**. Passwords are more secure if they also contain numbers and punctuation.

- [ ] Try entering a dictionary word. How long would it take a computer to guess that password?

Challenge: Creating a better password

Can you enter a password that would take a computer more than 1,000 years to crack but isn't too long to type?

Remember that your password is harder to guess if it's:

- [ ] Long
- [ ] Not a word in the dictionary
- [ ] Contains letters, numbers and punctuation

You're going to generate passwords that are hard for a computer to crack. These are useful for protecting important accounts. Note that many adults use a password manager program to help them remember lots of tricky passwords.

Step 2: Random characters

Let's create a program to choose a random character for your password.

Activity Checklist

- [ ] Open the blank Python template Trinket: **jumpto.cc/python-new**.
- [ ] Create a list of characters, stored in a variable called `chars`.

```
main.py
1  #!/bin/python3
2
3  chars = 'abcdefghijklmnopqrstuvwxyz'
```

- [ ] To choose a random character, you'll need to `import` the `random` module.

```
main.py
1  #!/bin/python3
2  import random
3
4  chars = 'abcdefghijklmnopqrstuvwxyz'
```

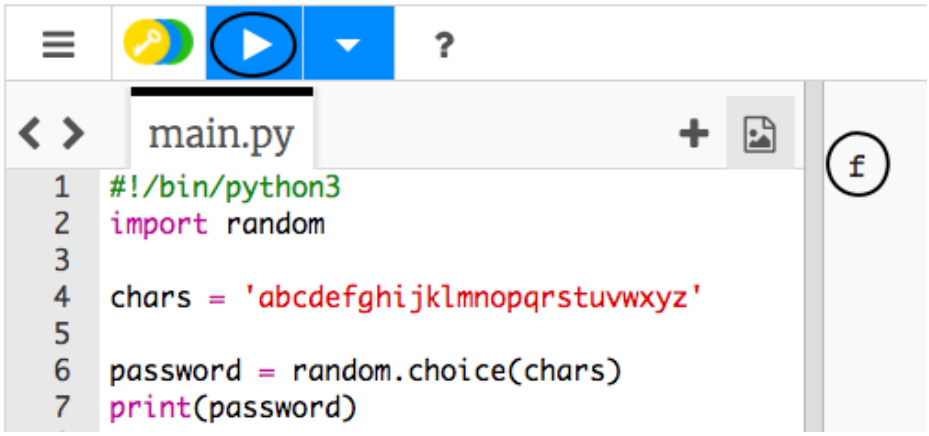
- [ ] Now you can choose a random character from the list, and store it in a variable called `password`.

```
main.py
1  #!/bin/python3
2  import random
3
4  chars = 'abcdefghijklmnopqrstuvwxyz'
5
6  password = random.choice(chars)
```

- [ ] Finally, you can print your (very short!) password to the screen.

```
main.py
1  #!/bin/python3
2  import random
3
4  chars = 'abcdefghijklmnopqrstuvwxyz'
5
6  password = random.choice(chars)
7  print(password)
```

- [ ] Test your project by clicking ‘run’. You should see a single random character on the screen.



If you run your program a few times, you should see different characters appear.

- [ ] A password isn’t very secure if it only contains letters. Add some numbers to your `chars` variable.

```
import random

chars = 'abcdefghijklmnopqrstuvwxyz1234567890'
```

- [ ] Test your code again a few times, and you should see that sometimes a number is chosen.

Save your project

Challenge: Using numbers and punctuation

Can you improve your program, so that it also chooses from:

- [ ] Capital letters (A-Z)
- [ ] Numbers (0-9)
- [ ] Punctuation (!?.,-)

You'll need to add to your `chars` variable. Remember to test your improved program!

Save your project

Step 3: A random password

A single character isn't very useful - let's improve your program to create a longer password.

Activity Checklist

- [ ] To create a password, you will add random characters to it, one at a time.

  To start with, your `password` variable should be empty. Add this line to your code:

```
main.py
1  #!/bin/python3
2  import random
3
4  chars = 'abcdefghijklmnopqrstuvwxyzABCDEFGHI.
5
6  password = ''
7  password = random.choice(chars)
8  print(password)
```

- [ ] You want to choose a random character 10 times. To do this, add the following code:

```
4  chars = 'abcdefghijklmnopqrstuvwxy
5
6  password = ''
7  for c in range(10):
8  password = random.choice(chars)
9  print(password)
```

- [ ] You should also indent (move in) the line to choose a random character, so that it happens 10 times.

  To indent, press the 'tab' key.

```
5
6  password = ''
7  for c in range(10):
8      password = random.choice(chars)
9  print(password)
```

- [ ] You need to use `+=` to **add** the new character to the password each time.

```
5
6  password = ''
7  for c in range(10):
8    password += random.choice(chars)
9  print(password)
```

- [ ] Test your new code and you should see a password that's 10 characters long.

```
VbP?S£(rkz
```

Save your project

Challenge: A longer password

Can you change your program so that it creates a verrrrrrrry long password?

```
ueHKO&w@kz$p4v8@JKR2rBJU@Dfji?H4),w*SQ*&
```

Save your project

Step 4: Choosing a password length

Some websites require passwords to be a certain length. Let's allow the user to choose the length of their password.

Activity Checklist

- [ ] First, ask the user to input a password length, and store it in a variable called `length`.

```
3
4  chars = 'abcdefghijklmnopqrstuvwxyzAB
5
6  length = input('password length?')
7
8  password = ''
```

- [ ] Use `int()` to turn the user's input into a whole number.

```
4  chars = 'abcdefghijklmnopqrstuvwxyzAB
5
6  length = input('password length?')
7  length = int(length)
8
9  password = ''
```

- [ ] Use your `length` variable to repeat as many times as the user entered.

```
6  length = input('password length?')
7  length = int(length)
8
9  password = ''
10 for c in range(length):
11   password += random.choice(chars)
12 print(password)
```

- [ ] Test your code. The password created should be the length entered by the user.

```
password length? 25
yyvuhKjF7&Fc?r7^hBI$XjRj5

password length? 5
A9b0V
```

Save your project

Step 5: Lots of passwords

Let's allow the user to create 3 passwords at once.

Activity Checklist

- [ ] Add this code to create 3 passwords:

```
5
6  length = input('password length?')
7  length = int(length)
8
9  for p in range(3):
10 password = ''
11 for c in range(length):
12   password += random.choice(chars)
13 print(password)
```

- [ ] Highlight the code for creating a password, and press tab to indent so that it repeats 3 times.

```
7  length = int(length)
8
9  for p in range(3):
10   password = ''
11   for c in range(length):
12     password += random.choice(chars)
13   print(password)
```

- [ ] Test your new code. You should now see 3 passwords of your chosen password length.

```
password length? 10
!!xgFu£*v5
v2E2,D(X1Z
0,MP£WXU£X
```

Save your project

Challenge: Choosing the number of passwords

Instead of always printing 3 passwords, can you allow the user to enter the number of passwords they want?

Here's how your program should work:

```
number of passwords? 4
password length? 15
qEIz2CCQpJq9,xF
^cZICdXSFAREHW7
e@4G*%176,£@HCm
x5seh^YX7N6o8CP
```

The code you'll need is **very** similar to the code for entering the `length` of the password.

Save your project