

Password

Learning Goals

- Practice iterating through a string
- Practice using the `ctype` library
- Practice using Boolean variables

Background

As we all know by now, it's important to use passwords that are not easy to guess! Many web apps now require passwords that require not only alphabetical characters, but also number and symbols.

In this lab, the user is prompted for a password, which will then be validated using a function `check` that you will complete. If the password contains at least one upper case letter, one lower case letter, a number and a symbol (meaning a printable character that's not a letter or number, such as '!', '\$' and '#' the function should return `true`. If not it should return `false`.

- Hints
 1. The `ctype` library has many useful functions that can come in handy here.
 2. Boolean variables can keep track of whether each criteria in an algorithm is valid.

Demo

```
password/ $ █
```

Getting Started

1. Log into cs50.dev using your GitHub account.
2. Click inside the terminal window and execute `cd`.
3. At the `$` prompt, type `mkdir password`
4. Now execute `cd password`
5. ~~Then copy and paste `wget https://cdn.cs50.net/2022/fall/labs/2/password.c` into your terminal to download this lab's distribution code.~~ Create a `password.c` file and write the thing.
6. You are to complete the function, `check`, which returns `true` if the password passes all criteria, and `false` if it does not. (and then actually in `main` use that info)

Implementation Details

Your function will iterate through the password that's supplied to it as an argument. Since you have to find at least one lower case letter, one upper case letter, one number and one symbol, you may want to create a

boolean variable for each and set each to `false` before you iterate through the string. If you then find a number, for instance you can set that boolean to `true`. If all booleans are `true` at the end of the function, it means all criteria are met, and you would return `true`.

Thought Question

- How many different passwords do you think can be made that are 8 letters long, can use any of 95 printable ASCII characters?

How to Test Your Code

Your program should behave per the examples below.

```
password/ $ ./password
Enter your password: hello
Your password needs at least one uppercase letter, lowercase letter, number and symbol!
```

```
password/ $ ./password
Enter your password: h3ll(
Your password needs at least one uppercase letter, lowercase letter, number and symbol!
```

```
password/ $ ./password
Enter your password: h3LL0!
Your password is valid!
```

You can check your code using `check50`, a program that CS50 will use to test your code when you submit, by typing in the following at the `$` prompt. But be sure to test it yourself as well!

```
check50 cs50/labs/2022/fall/password
```

To evaluate that the style of your code, type in the following at the `$` prompt.

```
style50 password.c
```

How to Submit

No need to submit! This is an optional practice problem completed with your lab.