Week 1 - Project Strong Password Generator

Jonathon, Vincent, Thao

Agenda

- Description of game
- Process
- Challenges
- Highlights
- Demo (ONE MINUTE)

1. Description

Generate a strong password:

➔ **Definition:**

Hard to guess/remember

Takes long for hackers to crack

➔ **Requirements**

Includes [a-zA-Z0-9$%%^]

Has certain length (e.g. 20 characters)

IRON HACK

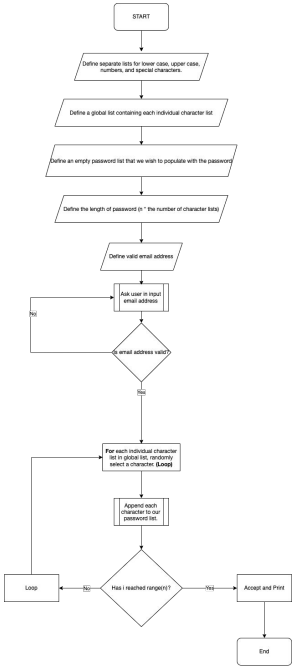
2. Process

➔ Pseudocode

- ➔ Define separate lists for lower case, upper case, numbers, and special characters.
- ➔ Define a global list containing each individual character list
- ➔ Define an empty password list that we wish to populate with the password
- ➔ Define the length of password (n * the number of character lists)
- ➔ **Define function**
  - ◆ **For** each individual character list in global list, randomly select a character. **(Loop)**
  - ◆ **Append** each character to our password list.
  - ◆ **For** n times repeat this process **(Loop)**
  - ◆ **End** when length is satisfied (20 char)
  - ◆ Print password
- ➔ **Validate** emails to generate password **(Loop)**

IRON HACK

2. Process


START
Define separate lists for lower case, upper case, numbers, and special characters.
Define a global list containing each individual character list
Define an empty password list that we wish to populate with the password
Define the length of password (= * the number of character lists)
Define valid email address
Ask user in input email address
No
Is email address valid?
Yes
For each individual character list in global list, randomly select a character. (Loop)
Append each character to our password list.
Loop
No
Has i reached range(n)?
Yes
Accept and Print
End

➔ Python Code

```
import random
import re

def generate_password():
  character_number = [chr(i) for i in range(48,57)]
  character_symbol = [chr(i) for i in range(33,47)]
  character_lower = [chr(i) for i in range(97,122)]
  character_upper = [chr(i) for i in range(65,90)]
  total_character = [chr(i) for i in range(33,122)]
  lists = []
  lists.append(character_number)
  lists.append(character_symbol)
  lists.append(character_lower)
  lists.append(character_upper)

  password = []
  n = 5

  for i in range(n):
    for j in range(4):
      randomItem = random.sample(lists[j],1)
      password.append(randomItem)

  password_str = ''.join(map(str,password))
  password_str = ''.join([i[0] for i in password])
  password_str = ''.join(random.sample(password_str,len(password_str)))
  return ("Your Password is: %s"%password_str)
generate_password()

username = str(input("Your email is: "))
pattern = "[\w+\.-]+@[\w+]+\.[\w+]+"

while True:

  if re.findall(pattern, username):
    print(generate_password())
    break
  else:
    print("Please input a proper email address:")
    type2 = str(input("Your email is: "))
    username = type2
    generate_password()
```

```
Your email is: rfeefhg@kjdfhjdf.com
Your Password is: Bx#B604R0'VH1*s'q4e,
```

IRON HACK

3. Challenges

➔ **What**

Logic:

- How to randomize with conditions

Technical:

- Type (list, str operations)
- Loop (for/ while, double)
- Jira navigation/ Git management

➔ **How We Tackle Them**

- Lab Material/Slides
- Google ⇒ Test ⇒ Google
- Human google (a.k.a - TAs)

IRON HACK

4. Highlight

➔ **Code Discovery**
When codes work (even when we don't know why)

➔ **Jira/ Git Enlightenment**
Some learning along the way

➔ **Finish**
Entire code works after a few small roadblocks

IRON HACK

DEMO

https://colab.research.google.com/drive/13-Y8OR8C5RI0tL2XF_6dE_v8_ECMMf7d#scrollTo=4Ns8fZcWxh1H