Back

Practice Quiz: Basic Regular Expressions

Practice Quiz • 12 min • 6 total points

Congratulations! You passed!

Grade received 100%   **To pass** 80% or higher

Go to next item

1. The check_web_address function checks if the text passed qualifies as a top-level web address, meaning that it contains alphanumeric characters (which includes letters, numbers, and underscores), as well as periods, dashes, and a plus sign, followed by a period and a character-only top-level domain such as ".com", ".info", ".edu", etc. Fill in the regular expression to do that, using escape characters, wildcards, repetition qualifiers, beginning and end-of-line characters, and character classes.

1 / 1 point

```python
import re
def check_web_address(text):
  pattern = r"^([a-zA-Z0-9]+([\-\+\._]?[a-zA-Z0-9]+)*)\.([a-zA-Z]{2,})$"
  result = re.search(pattern, text)
  return result != None

print(check_web_address("gmail.com")) # True
print(check_web_address("www@google")) # False
print(check_web_address("www.Coursera.org")) # True
print(check_web_address("web-address.com/homepage")) # False
print(check_web_address("My_Favorite-Blog.US")) # True

```

Run

Reset

```
True
False
True
False
True
```

Correct

Right on! No bogus web address will get past you!

2. The check_time function checks for the time format of a 12-hour clock, as follows: the hour is between 1 and 12, with no leading zero, followed by a colon, then minutes between 00 and 59, then an optional space, and then AM or PM, in upper or lower case. Fill in the regular expression to do that. How many of the concepts that you just learned can you use here?

1 / 1 point

```python
import re
def check_time(text):
  pattern = r"^(1[0-2]|0?[1-9]):([0-5][0-9])(\s?(AM|am|PM|pm))?$"
  result = re.search(pattern, text)
  return result != None

print(check_time("12:45pm")) # True
print(check_time("9:59 AM")) # True
print(check_time("6:60am")) # False
print(check_time("five o'clock")) # False
```

Run

Reset

```
True
True
False
False
```

Correct

You nailed it! It's "time" to celebrate!

3. The contains_acronym function checks the text for the presence of 2 or more characters or digits surrounded by parentheses, with at least the first character in uppercase (if it's a letter), returning True if the condition is met, or False otherwise. For example, "Instant messaging (IM) is a set of communication technologies used for text-based communication" should return True since (IM) satisfies the match conditions." Fill in the regular expression in this function:

1 / 1 point

```python
import re
def contains_acronym(text):
  pattern = r"\([A-Z0-9]?[A-Za-z]{2,}\)"
  result = re.search(pattern, text)
  return result != None

```

```
print(contains_acronym("Instant messaging (IM) is a set of communication technologies used for text-based communication"))
print(contains_acronym("American Standard Code for Information Interchange (ASCII) is a character encoding standard for el
print(contains_acronym("Please do NOT enter without permission!")) # False
print(contains_acronym("PostScript is a fourth-generation programming language (4GL)")) # True
print(contains_acronym("Have fun using a self-contained underwater breathing apparatus (Scuba)!")) # True
```

Run

Reset

```
True
True
False
True
True
```

Correct

Great job! Now that's real Personal Responsibility In
Delivering Excellence (PRIDE)!

4. What does the "r" before the pattern string in re.search(r"Py.*n", sample.txt) indicate?

1 / 1 point

- [✓ Raw strings]
- Regex
- Repeat
- Result

Correct

Right on! "Raw" strings just means the Python interpreter won't try to interpret any special characters and, instead, will just pass the string to the function as it is.

5. What does the plus character **[+]** do in regex?

1 / 1 point

- Matches plus sign characters
- [✓ Matches one or more occurrences of the character before it]
- Matches the end of a string
- Matches the character before the **[+]** only if there is more than one

Correct

Awesome! The plus character [+], matches ***one or more*** occurrences of the character that comes before it.

6. Fill in the code to check if the text passed includes a possible U.S. zip code, formatted as follows: exactly 5 digits, and sometimes, but not always, followed by a dash with 4 more digits. The zip code needs to be preceded by at least one space, and cannot be at the start of the text.

1 / 1 point

```
import re
def check_zip_code (text):
  result = re.search(r"\s\d{5}(-\d{4})?", text)
  return result != None

print(check_zip_code("The zip codes for New York are 10001 thru 11104.")) # True
print(check_zip_code("90210 is a TV show")) # False
print(check_zip_code("Their address is: 123 Main Street, Anytown, AZ 85258-0001.")) # True
print(check_zip_code("The Parliament of Canada is at 111 Wellington St, Ottawa, ON K1A0A9.")) # False
```

Run

Reset

```
True
False
True
False
```

Correct

Woohoo! You're zipping through these regular expressions
like a champ!