

Question 1

Quelle regex correspond à une adresse email valide ?

A)

`^[a-zA-Z0-9_+.-]+@[a-zA-Z0-9-]+\.[a-zA-Z0-9-.] +$`

B) `^[\\w]+@[\\w]+\\. [\\w]+$`

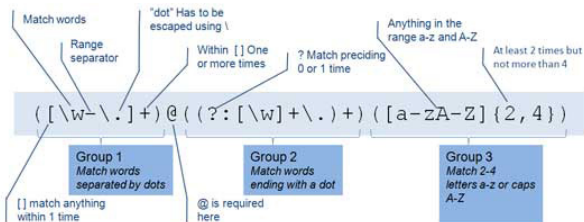
C) `^\\w+@\\w+\\. \\w{2,3}$`

D) `^[a-z0-9._%+.-]+@[a-z0-9.-]+\\. [a-z]{2,}$`

Réponse:

1. A -

`^[a-zA-Z0-9_+-.]+@[a-zA-Z0-9-]+\.[a-zA-Z0-9-\.]+$`



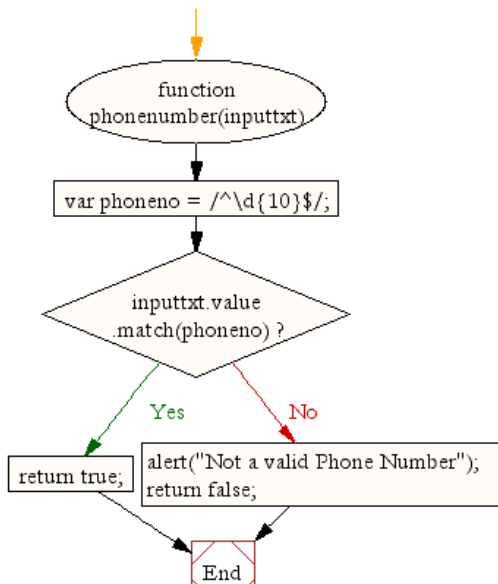
Question 2

Quelle regex correspond à un numéro de téléphone au format (123) 456-7890 ?

- A) `^\d{3} \d{3}-\d{4}$`
- B) `^\((\d{3})\ \d{3}-\d{4}$`
- C) `^\(\d{3}) \d{3}-\d{4}$`
- D) `^\((\d{3})\ \d{3}\-\d{4}$`

Réponse:

2. B - `^\(\d{3}\) \d{3}-\d{4}$`



Question 3

Quelle regex correspond à une chaîne qui commence par une lettre majuscule suivie de n'importe quel nombre de lettres minuscules ?

- A) $\wedge[A-Z][a-z]*\$$
- B) $\wedge[A-Z]+[a-z]\$$
- C) $\wedge[A-Z][a-z]+\$$
- D) $\wedge[A-Z]\backslash w+\$$

Réponse:

3. **A** - $\wedge[A-Z][a-z]*\$$



Capitalize first letter of every word

```
1 const str = "code drops";
2
3 const capitalize = (input) =>
4   input
5     .split(" ") // split by space
6     .map((word) => word[0].toUpperCase() + word.slice(1)) // uppercase
7     // first character of every word and concatenate with rest of the string
8     .join(" "); // join them back by space
9   capitalize(str); // Code Drops
```

JAVASCRIPT

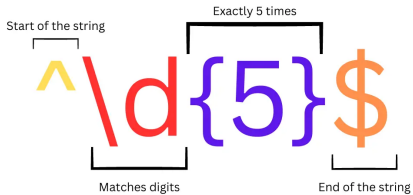
Question 4

Quelle regex correspond à une chaîne qui ne contient que des chiffres ?

- A) `^\d*$`
- B) `^\d+$`
- C) `^[0-9]+$`
- D) `^\d{1,}$`

Réponse:

4. B - `^\d+$`



Question 5

Quelle regex correspond à une date au format YYYY-MM-DD ?

- A) `^\d{4}-\d{2}-\d{2}$`
- B) `^\d{4}/\d{2}/\d{2}$`
- C) `^\d{2}-\d{2}-\d{4}$`
- D) `^\d{4}-\d{1,2}-\d{1,2}$`

Réponse:

5. A - `^\d{4}-\d{2}-\d{2}$`

REGULAR EXPRESSION v1 ▾ 2 matches, 51 steps (~1ms)

`/((([0-9]{4})-([0-9]{2})-([0-9]{2})))|([0-9]{4})\v([0-9]{2})\v([0-9]{2}))/gm`

TEST STRING SWITCH TO UNIT TESTS ▸

2019/12/13
2019-12-13

SUBSTITUTION ▾

`\1\5`

2019/12/13
2019-12-13

Question 6

Quelle regex correspond à un mot qui commence et se termine par une voyelle (a, e, i, o, u) ?

- A) `^[aeiou].*[aeiou]$`
- B) `^[aeiou].*[aeiou]`
- C) `^\w*[aeiou]\w*$`
- D) `^\w*[aeiou].*[aeiou]\w*$`

Réponse:

6. A - `^[aeiou].*[aeiou]$`

Expression	sha
<code>/\w*([aeiou]{2})\w*/gm</code>	
Text	
Words that contain 2 or more 2 letter sequences of vowels:	
visionproof, steamier, preequip	

Question 7

Quelle regex correspond à une chaîne de caractères alphanumériques de longueur exactement 8 ?

- A) `^[a-zA-Z0-9]{8}$`
- B) `^\w{8}$`
- C) `^[0-9a-zA-Z]{8}$`
- D) Toutes les réponses ci-dessus.

Réponse:

7. D - Toutes les réponses ci-dessus.

Anchor	Description
<code>[:lower:]</code>	lower-case letters
<code>[:upper:]</code>	upper-case letters
<code>[:alpha:]</code>	alphabetic characters <code>[:lower:] + [:upper:]</code>
<code>[:digit:]</code>	numeric values
<code>[:alnum:]</code>	alphanumeric characters <code>[:alpha:] + [:digit:]</code>
<code>[:blank:]</code>	blank characters (space & tab)
<code>[:cntrl:]</code>	control characters
<code>[:punct:]</code>	punctuation characters: ! " # % & ' () * + , - . / : ;
<code>[:space:]</code>	space characters: tab, newline, vertical tab, space, etc
<code>[:xdigit:]</code>	hexadecimal digits: 0-9 A B C D E F a b c d e f
<code>[:print:]</code>	printable characters <code>[:alpha:] + [:punct:] + space</code>
<code>[:graph:]</code>	graphical characters <code>[:alpha:] + [:punct:]</code>

*adapted from *Handling and Processing Strings in R* (Sanchez, 2013)

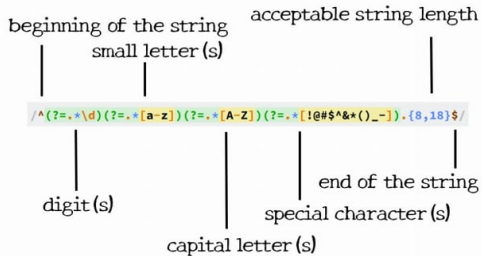
Question 8

Quelle regex correspond à un mot qui ne contient que des lettres majuscules ou minuscules, mais pas de chiffres ?

- A) `^[a-zA-Z]*$`
- B) `^[a-zA-Z]+$`
- C) `^\d+$`
- D) `^[A-Za-z]+$`

Réponse:

8. A - $\text{^[a-zA-Z]*\$}$



Question 9

Quelle regex correspond à une chaîne qui ne contient pas la lettre “a” ?

- A) `^[^a]*$`
- B) `^[^a]+$`
- C) `^[a-zA-Z]*$`
- D) `^.*[^a].*$`

Réponse:

9. A - $^{\wedge}[^a]*\$$

B5	✕	✓	<i>fx</i>	=RegExpMatch(A5, \$A\$2)
	A	B		
1	Pattern			
2	$^{\wedge}[^a]*\$$			
3				
4	Phone number	Country code missing		
5	+1 123.237.4567	FALSE		
6	(541) 754-3010	TRUE		
7	(123).345.4567	TRUE		
8	+44 7911 123456	FALSE		
9	+375 (447) 381298	FALSE		

Question 10

Quelle regex correspond à une adresse IP au format IPv4 (ex : 192.168.1.1) ?

- A) `^\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}$`
- B) `^([0-9]{1,3}\.){3}[0-9]{1,3}$`
- C) `^(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)$`
- D) Toutes les réponses ci-dessus.

Réponse:

10. **C** - $\wedge(25[0-5] | 2[0-4][0-9] | [01]?[0-9][0-9]?)\backslash.(25[0-5] | 2[0-4][0-9] | [01]?[0-9][0-9]?)\backslash.(25[0-5] | 2[0-4][0-9] | [01]?[0-9][0-9]?)\backslash.(25[0-5] | 2[0-4][0-9] | [01]?[0-9][0-9]?)\$$

```
def validate_ip_regex(ip_address):
    match = re.match(
        r"[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}", "157.36.24.19", ip_address)

    if not bool(match):
        print(f"The IP address {ip_address} is not valid")
        return False

    bytes = ip_address.split(".")

    for ip_byte in bytes:
        if int(ip_byte) < 0 or int(ip_byte) > 255:
            print(f"The IP address {ip_address} is not valid")
            return False

    print(f"The IP address {ip_address} is valid")
    return True
```