

Grupo 1 - Interpretação

Considere as funções JavaScript (JS) definidas na última folha do enunciado.

Considere que o valor da constante `MEU_NUM_DE_ALUNO` é uma frase correspondente ao seu número de aluno e utilize esse valor em sua substituição. Por exemplo, se o seu número de aluno for 321, considere que a constante vale "321" e responda, onde for relevante, em função desse valor.

1. Qual o valor de `result_f1` depois de executar-se o código seguinte?

```
var result_f1 = f1(MEU_NUM_DE_ALUNO)
document.write(result_f1)
document.write("<hr>")
```

2. Qual o valor de `result_f2` depois de executar-se o código seguinte?

```
var result_f2 = f2(MEU_NUM_DE_ALUNO)
```

3. Qual o valor de `result_f3` depois de executar-se o código seguinte?

```
var result_f3 = f3(MEU_NUM_DE_ALUNO)
```

4. Qual o valor de `result_f4` depois de executar-se o código seguinte?

```
var result_f4 = f4(MEU_NUM_DE_ALUNO)
```

5. Qual o valor de `result_f5` depois de executar-se o código seguinte?

```
var MEU_OBJETO = {
  "num":123,
  "nome":"anon",
  "b":false,
  "d":new Date().getFullYear()
}
var result_f5 = f5(MEU_OBJETO)
```

6. Qual o valor de `result_f6` depois de executar-se o código seguinte?

```
var result_f6 = f6(6)
```

7. Qual o valor de `result_f7` depois de executar-se o código seguinte?

```
var result_f7 = f7()
```

8. Qual o valor de `result_f8` depois de executar-se o código seguinte?

```
var result_f8 = f8([1,1,1])
```

9. Considere a seguinte estrutura HTML e responda, nesse contexto, como fica a estrutura, depois de executar-se a função `f9`, conforme chamada de seguida.

```
<html><body><p id="idP"></p></body></html>
f9("idP")
```

10. Qual o valor de `result_f10` depois de executar-se o código seguinte?

```
const X = ["top", "fun", "happy", "easy", "hard", "ok", "okay"]
var result_f10 = f10()
```

11. Qual o valor de `result_f11` depois de executar-se o código seguinte?

```
const Y = ['g', 'o', 'o', 'd']
var result_f11 = f11()
```

Grupo 2 - HTML + JS + Python Flask

Year and month: _____

Year:

Month:

```
[{"mDrawNumber": 1311, "mYear":
"mBalls": [33, 17, 3, 39, 10], "mStar
14, 33, 48], "mStarsByDrawnOrder"
```

12. Escreva, em **HTML**, uma página que, quando apresentada, tenha o aspeto sugerido pela imagem acima.

A página contém uma form e uma section para "feedback".

A form, com id "**idForm**", aceita apenas o submit de dois inputs numéricos:

- um para escrita de um ano, com o id "**idYear**";
- um para escrita de um mês, com o id "**idMonth**".

A ideia é submeter pedidos a um recurso que produz respostas JSON a serem escritas na section de feedback, que tem id "**idFeedback**".

Note que tudo o que está escrito na section de feedback é apenas exemplificativo e não faz parte da estrutura que é pedida nesta questão.

13. Escreva, em **JS**, o comportamento da página **HTML** da questão anterior, satisfazendo os seguintes requisitos:

R1 - utilize o padrão e as boas práticas que estudou;

R2 - só são aceitáveis números nos inputs;

R3 - o input do ano tem que ter exatamente 4 símbolos;

R4 - o input do mês pode ter 1 ou 2 símbolos;

R5 - se R2, R3 ou R4 não forem satisfeitos, não será possível submeter a form, e deverá obter-se um alerta disso;

R6 - quando a form for submetida, deve acontecer um pedido assíncrono, por GET, a um recurso remoto, em endereço com a seguinte forma:

```
http://localhost:5005/results/<year>/<month>
```

Year e month deverão ser os valores concretos entrados pelo utilizador; por exemplo 2024 e 5, situação em que o URL seria `http://localhost:5005/results/2024/5`

R7 - antes de submeter o pedido, deve aparecer uma janela de alert a informar o endereço que vai ser solicitado;

R8 - quando o pedido for respondido, a resposta deve ficar disponível na zona de feedback.

14. Escreva uma app **Python Flask** que saiba responder apenas a pedidos **GET** com a seguinte estrutura:
http://<domain name>:<port>/results/<year>/<month>

por exemplo: http://localhost:5005/results/2020/2

Assuma que esta app tem acesso a uma base dados de todos os resultados alguma vez sorteados no jogo de azar "EuroMillions".

A base de dados está disponível num ficheiro de texto "**BD.JSON**", em formato JSON, na pasta static. O formato do ficheiro é exatamente o formato das respostas que têm que ser produzidas, exemplificado adiante.

Os URLs permitem obter respostas de texto, estruturadas em JSON. Cada resposta é uma lista dos sorteios que têm o ano e o mês indicados pelos inputs da form.

Segue-se um exemplo da resposta para ano de 2020, mês 2, limitada a um só sorteio.

Como se observa, cada resposta é um dicionário com chaves **mDrawNumber**, **mYear**, **mMonth**, **mDay**, **mBalls** e **mStars**.

Para este exercício, só é importante compreender que a chave **mYear** representa o ano do sorteio; e a chave **mMonth** o mês do sorteio.

```
[  
{ "mDrawNumber": 1298, "mYear": 2020, "mMonth": 2, "mDay": 28, "mBalls": [8, 11, 23, 20, 22], "mStars": [4, 3] },  
... ]
```

A sua app:

- só deve suportar a route indicada;
- tem que saber ler o ano e mês solicitados;
- tem de produzir uma resposta correta, confirme descrito.

```

function odd_numbers_in_range(p_min, p_max) {
    ret = []
    for (var n=p_min; n<=p_max; n+=1) {
        if (n%2!=0) ret.push(n)
    }
    return ret
} //odd_numbers_in_range

function f1 (p_num) {
    var ns = "" + p_num // the p_num as string
    var idxs = odd_numbers_in_range(0, p_num.length - 1)
    for (var idx = 0, ret=""; idx < idxs.length; idx += 1)
        ret += ns.charAt(idxs[idx])
    return ret
} //f1

function f2 (p) { return typeof(p) }

function f3 (p) { return p.split("").sort() }

function aux_f4(a, b) {
    if(a>b) return -1 ; if(a<b) return 1 ; return 0 ;
}
function f4 (p) { return p.split("").sort(aux_f4) }

function f5(p_object) {
    var ret = ""
    for (var prop in p_object) {
        var v = p_object[prop]
        if(!p_object[prop])
            ret+=v.toString()
    } //for
    return ret
} //f5

function f6(p_limite=100) { return p_limite==0 ? 0 :
p_limite+f6(p_limite-2) } //f6

```

```

function f7(p_uma_data) {
  p_uma_data = p_uma_data===undefined ? new Date() : p_uma_data
  return {
    "r": p_uma_data.getMonth() < 6 ? "<6" : ">=7",
    "m": p_uma_data.getDate() >= 15 ? "sim" : "não",
    "f": p_uma_data.getDay() == 6 ? "favorito" : "outro"
  }
} //f7

```

```

function f8(p_col) {
  var soma = media = 0
  for (var v of p_col)
    soma += v
  media = soma / p_col.length
  return soma + media + ((soma + media) % 2)
} //f8

```

```

function f9(p_html_id) {
  var p = document.getElementById(p_html_id)
  if (p != null) {
    var ctrl = [2, 1]
    while (ctrl.length > 0) {
      p.innerHTML += p.innerHTML + ctrl[0]
      ctrl.shift()
    }
  }
} //f9

```

```

function f10() {
  return X[new Date().getDay()]
} //f10

```

```

function f11(p=Y) {
  var ret = ""
  while (p.length) {
    ret += p.length % 2 === 0 ? p[0] : p.length + ""
    p.pop()
  }
  return ret
} //f11

```