

# مکتب شریف

اولین بوتکمپ آموزشی - استخدامی ایران



سری نهم

مکتب ۹۴





## ۱- بخش JavaScript

1-1- در هر کدام از فراخوانی های زیر، با ذکر دلیل نتیجه را مشخص کنید.

```
function test(num = 1) {  
  console.log(typeof num)  
}
```

```
test()  
test(undefined)  
test('')  
test(null)
```

1-2- خروجی قطعه کد زیر را مشخص کنید و سپس مشخص کنید چه روش هایی می توان به کار برد که خروجی حالت اول و سوم، یکسان باشد.

```
function PassbyReference(obj){  
  let tmp = obj.a;  
  obj.a = obj.b;  
  obj.b = tmp;  
  console.log(`Inside Pass By Reference  
    Function -> a = ${obj.a} b = ${obj.b}`);  
}  
let obj={a: 10, b: 20}  
console.log(`Before calling Pass By Reference  
  Function -> a = ${obj.a} b = ${obj.b}`);  
PassbyReference(obj)  
console.log(`After calling Pass By Reference  
  Function -> a = ${obj.a} b = ${obj.b}`);
```



3-1- یک تابع `spy (func)` ایجاد کنید که یک `wrapper` را برمی گرداند که همه فراخوانی ها را ذخیره می کند تا در پراپرتی `calls` خود اعمال کند.

هر فراخوانی به عنوان آرایه ای از آرگومان ها ذخیره می شود.

مثال:

```
function work(a, b) {
  alert( a + b ); // work is an arbitrary function or method
}

work = spy(work);

work(1, 2); // 3
work(4, 5); // 9

for (let args of work.calls) {
  alert( 'call:' + args.join() ); // "call:1,2", "call:4,5"
}
```

4-1- یک تابع `delay(f, ms)` بنویسید که هر فراخوانی تابع `f` را به اندازه `ms` به تاخیر می اندازد.

مثال:

```
function f(x) {
  alert(x);
}

// create wrappers
let f1000 = delay(f, 1000);
let f1500 = delay(f, 1500);

f1000("test"); // shows "test" after 1000ms
f1500("test"); // shows "test" after 1500ms
```

5-1- در نظر بگیرید یک متد `built-in` برای آرایه ها به صورت `arr.filter(f)` داریم که تمام عناصر را از طریق تابع `f` فیلتر می کند. اگر مقدار `true` را برگرداند، آن عنصر در آرایه به دست آمده برگردانده می شود.

مجموعه ای از فیلترهای آماده به شکل زیر بسازید.

- `inBetween(a, b)` – between a and b or equal to them (inclusively).
- `inArray([...])` – in the given array.



مثال:

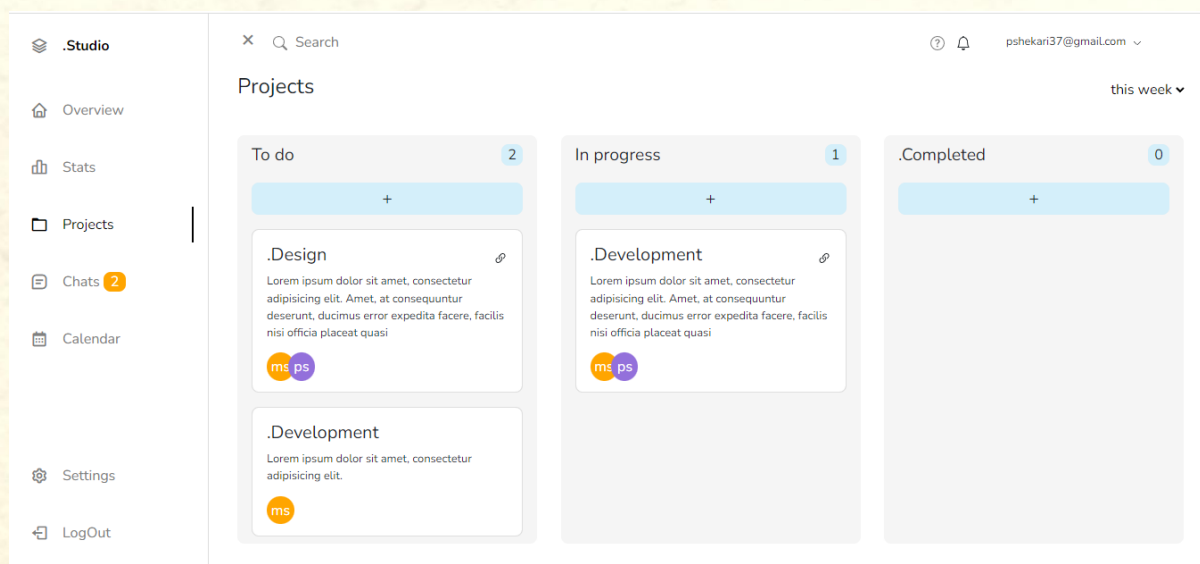
```
/* .. your code for inBetween and inArray */  
let arr = [1, 2, 3, 4, 5, 6, 7];  
  
alert( arr.filter(inBetween(3, 6)) ); // 3,4,5,6  
  
alert( arr.filter(inArray([1, 2, 10])) ); // 1,2
```

6-1- تابع جمع دو عدد را به گونه ای بنویسید که با شیوه فراخوانی زیر، عمل کند.

```
sum(1)(2) = 3  
sum(5)(-1) = 4
```

## ۲- بخش HTML و CSS

7-1- طرح زیر را در تمامی نماها ( دسکتاپ، موبایل و تبلت ) پیاده سازی کنید. ( با استفاده از tailwindcss )



### نکات

- مهلت ارسال تمرین تا پایان روز چهارشنبه ۱۴۰۲/۲/۶ است. پاسخ تمرین را در کارتابل آموزشی خود ارسال کنید.
- نام **فایل و ایمیل** ارسالی خود را به این صورت قرار دهید: **Name\_hw9\_Maktab94** به عنوان مثال **.Mohammad\_Ali\_Kargar\_hw9\_maktab94**
- تصاویر تمرین بخش HTML در کنار این فایل ضمیمه شده است.
- ارسال پاسخ تمرینی که در کلاس تحویل داده شده، الزامی است.
- در صورت لزوم یک فایل word به عنوان توضیح در کنار کدهای خود قرار دهید.



- در صورتی که تمرین شامل چند فایل و فولدر می باشد حتماً آن ها را قالب یک فایل فشرده شده تجمیع کنید.
- به دلیل بروز برخی مشکلات در ارسال فایل ها با پسوند **rar**، در صورتی که قصد ارسال فایل نهایی به صورت فایل **rar** دارید، پسوند آن را از **rar** به **ra** و در صورت عدم ارسال مجدد به **pdf** یا ... تغییر دهید.
- از ارسال شدن ایمیل خود اطمینان حاصل فرمایید.
- در صورتی که سوالی دارید **در گروه راکتچی** بپرسید.