

episode

setTimeout + closures interview question

```

- function x() {
  var i = 1;
  setTimeout(function () {
    console.log(i);
  }, 3000);
}
x();

```

⇒ after 3 seconds
value of i will
print i.e. 1

```

- function x() {
  var i = 1;
  setTimeout(function () {
    console.log(i);
  }, 3000);
  console.log("Hello");
}
x();

```

[JS doesn't
wait for anyone]



⇒ Hello
after 3 seconds
value of i will
print i.e. 1

[Hello]

using var

Page No.

Date

```
- function x() {  
  for (var i = 1; i <= 5; i++) {  
    setTimeout(function () {  
      console.log(i);  
    }, i * 1000);  
  }  
  console.log("Hello");  
}  
x();
```

[wrong way]



Hello

every 1 second

6

6

6

6

6

[due to
closures]

[How to fix
this]

using let

[correct way]

[let
has block
scope]

```
- function x() {  
  for (let i = 1; i <= 5; i++) {  
    setTimeout(function () {  
      console.log(i);  
    }, i * 1000);  
  }  
  console.log("Hello");  
}  
x();
```

Hello

every 1 second

1

2

3

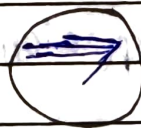
4

5

[everytime there
will be new
copy of i
due to let]

[don't use let but we want same output
as above, now what
to do]

```
- function x() {  
  for (var i = 1; i <= 5; i++) {  
    function close(b) {  
      setTimeout(function () {  
        console.log(b);  
      }, b * 1000);  
    }  
    close(i);  
  }  
}  
x();
```



1

2

3

4

5

[this works
because everytime we called close function
with i, it creates a new copy of i
for itself i.e. using close function
or we can say closure we kind of
created a new copy of i everytime setTimeout
was called]