**ES6**

Harmony === ES6 === ES2015

Q: debugger statement?

A: debugger;

**LET, CONST & VAR**

Hoisting:

\_Before JS code is executed, all variable (var) is hoisted to the top of the function scope (globally or locally to an entire functions scope)

Q: but for this piece of code

**function** **getClothing**(isCold) {

**if** (isCold) {

**var** freezing = 'Grab a jacket!';

} **else** {

**var** hot = 'It’s a shorts kind of day.';

console.log(freezing);

}

}

when ‘freezing’ and ‘hot’ are hoisted, do they carry their value (‘grab a jacket’) and ‘it’s shorts kind of day’ with them?

A: No, it’s only declared but not given a value at the beginning

Q: Why was that function’s console.log is undefined?

A: Because since it’s **false**, freezing is never assigned a value. Only when it’s true (or isCold ===true), then the value of freezing would be defined

Q: Let/ const vs Var

So if I use ‘let’ or ‘const inside a block of code (inside curly braces), the variables are trapped within the block they’re declared

Otherwise variable with var is hoisted up (declared only but not with the value)

DO NOT use var at all

Q: Let vs Const?

A: +)Variables declared with “let” can be reassigned but can’t be redeclared in the same scope

+) With Const, variables must be assigned an initial avalue but cannot be redeclared or reassigned

* Const is the strictest way to declare a variable

Ex:

Let friend = ‘Sarah’;

friend = ‘Maya’;

console.log(friend)

* Will be Maya

Q: So in this test, when to use “let”, and when to use “const”?

var CHARACTER\_LIMIT = 255;

var posts = [

"#DeepLearning transforms everything from self-driving cars to language translations. AND it's our new Nanodegree!",

"Within your first week of the VR Developer Nanodegree Program, you'll make your own virtual reality app",

"I just finished @udacity's Front-End Web Developer Nanodegree. Check it out!"

];

// prints posts to the console

function displayPosts() {

for (var i = 0; i < posts.length; i++) {

console.log(posts[i].slice(0, CHARACTER\_LIMIT));

}

}

displayPosts();

A:

**const** CHARACTER\_LIMIT = 255;

**const** posts = [

"#DeepLearning transforms everything from self-driving cars to language translations. AND it's our new Nanodegree!",

"Within your first week of the VR Developer Nanodegree Program, you'll make your own virtual reality app",

"I just finished @udacity's Front-End Web Developer Nanodegree. Check it out!"

];

// prints posts to the console

function displayPosts() {

for (**let** i = 0; i < posts.length; i++) {

console.log(posts[i].slice(0, CHARACTER\_LIMIT));

}

}

displayPosts();

* ***Const*** the first 2 because you don’t want to reassign or redeclare
* ***Let*** the last one because while you don’t want to redeclare, you want to reassign every time you go through the loop. NOTE: for the for loop, from now on, use ‘let’ and not ‘var’

Q:. slice: what to include, and what not?

A: slice(beginning, excluded\_last)

Ex: slice(0,3)

* Will yield the first, second, third but not the fourth item