Web Info Tech Revision

Week 2

Git

- Store files as snapshot commits, rather than delta
- Staging → Local → Remote

Command list

Commands	Meaning
git init	creates the repo
git clone	clones a remote repo
git status	status of repo
git add (-A/*.c)	add to staging area
git commit -m ""	commit changes to local
git push	push from local to remote
git log	shows commit history
git checkout <commit_id>	moves head to that commit
git diff	compares with last commit
git revert --no-commit <_id>	reverts to that commit
git branch	shows branches
git checkout -b <b_name>	creates a new branch
git merge <b_name>	merges with branch
git branch -d <b_name>	deletes branch

- Best practices
  - Commit often
  - Commit related changes together
  - Commit completed work
  - Branch before build
  - Commit with meaningful messages & naming
  - Agree on a workflow

Javascript

- Separation of concerns
  - Structure - HTML

    - Presentation - CSS
    - Behaviour - JS
    - Benefits: No duplication, increase maintainability, extensibility
- print to console:

```
console.log("Hello");
```

- change stuff:

```
<script>
function myFunction() {
  document.getElementById("demo").innerHTML = "Paragraph changed.";
}
</script>
```

- Include script

```
<script src="myScript.js"></script>
```

- String behaviour

```
var x = "5" + 2 + 3; // x = "523"
var y = 2 + 3 + "5"; // y = "55"
```

- var, const, let
    - var
        - scope: function
    - const
        - scope: block
        - must be assigned when defined and cannot change later
        - won't overwrite var
        - can change property in const object
    - let
        - scope: block
        - won't overwrite var
- Recommendation
    - Declare all variables at the top
    - Declare all functions before calling them

    - Prefer let and const
- JavaScript types are dynamic
- Anonymous function

```
setInterval(function(){
  console.log("1 Second"):
},1000);
```

- Array iteration

```
<button onclick="numbers.forEach(myFunction)">Try it</button>
<p>Sum of numbers in array: <span id="demo"> </span></p>

<script>
var sum = 0;
var numbers = [65, 44, 12, 4];
function myFunction(item) {
  sum += item;
  demo.innerHTML = sum;
}
</script>
```

- for loop

```
var array = [23, 34, 45, 56];
for (let e in array) // 0, 1, 2, 3
for (let e of array) // 23, 34, 45, 56
```

- DOM: Document Object Model
    - Accessing a function without () will return the function definition instead of the function result

    ```
    <p id="demo"></p>
    <script>
    function toCelsius(f) {
      return (5/9) * (f-32);
    }
    document.getElementById("demo").innerHTML = toCelsius;
    </script>
    ```

    - Retrieving elements: CSS selector

```
var h1 = document.querySelector(“h1”); // only the first
h1.style.color = “red”;

var h1 = document.querySelectorAll(“.class”); // returns node list

// adding elements
var newDiv = document.createElement("div");
document.body.appendChild(newDiv);
```