Code Standardization

```
var group2 = [
            “Diana",
            “Marieh",
            “Tushabe“,
            “Rino”,
             “Kakuru”,
             “Asega”,
             “Nickson”,
             “Camillus”
];
```

Code Standardization

Refers to developers following a recognizable pattern and establishing a standard in coding.

Area unit rules followed by developers so that programming language nuances stay constant.

Having multiple developers acquainted with one code base while following a recognizable pattern and standard code

Why is this so?

So that any new learner of the language will simply inform with and realize it and this further ensures that any developer WHO appears at can comprehend code and know what to expect through the complete application.

Examples being ASCII text file

In a nutshell, code standardization is following a general style/rule of coding.

Importance of Code Standardization

If the coding standards are NOT defined, developers can use any of their own methods, which may lead to:

01

Security Concerns

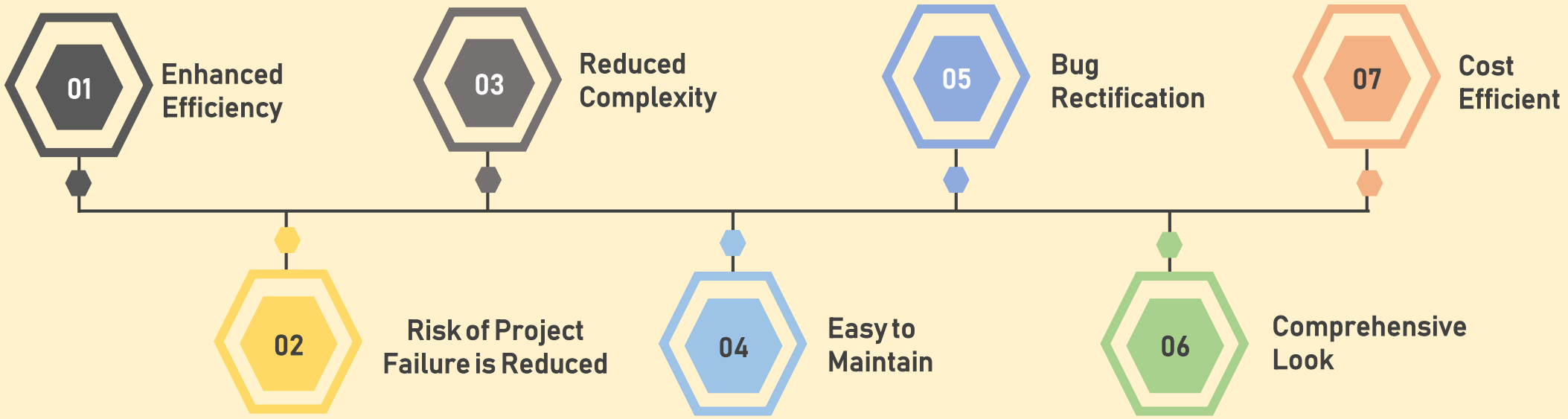
Software becomes vulnerable to attacks if it is inconsistent, contains bugs and errors in logic. Most of these problems are caused by faulty code due to poor coding practices.

02

Performance Issues

Poor coding can lead to performance issues like trouble when the user is interacting with the site, server response issues, reusability & flow of the code, etc.

Advantages of Coding Standards



```
const app = express();
const upload = multer();
// Enable CORS (see https://enable-cors.org)
app.use((req, res, next) => {
 res.header("Access-Control-Allow-Origin", "*");
 res.header(...);
```

Code Standardization in JavaScript (JS)

- Develop code that can be understood
- Don't yield to browser whims
- Stick to a strict coding style
- Build on shoulders of giants(use JS libraries)
- Keep DOM access to a minimum
- Call Things by there name
- Modularize- One function per task
- Avoid mixing with other technologies
- Comment as much as needed but not more

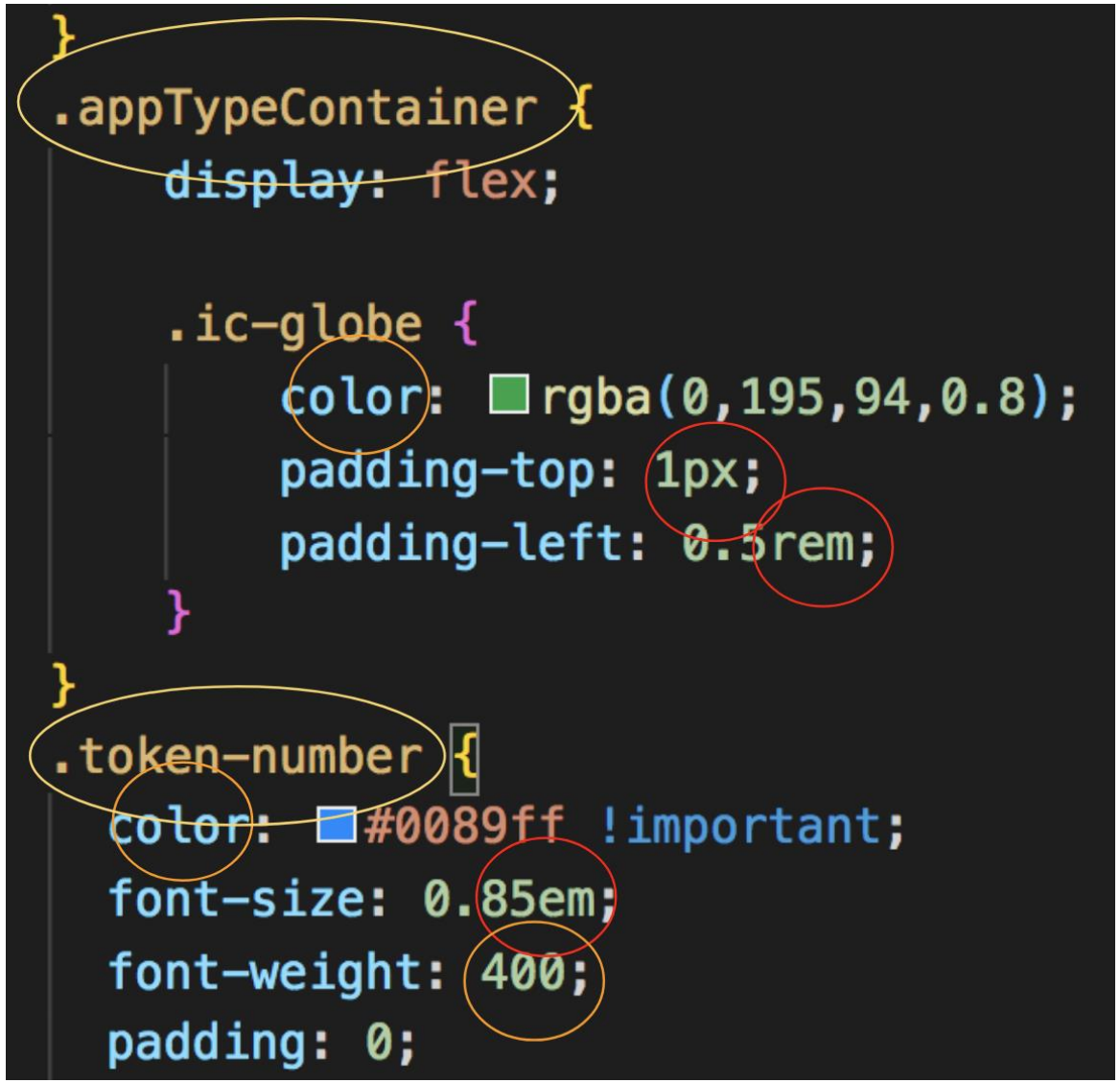
Airbnb JavaScript (JS) Code Standards

Style guide/ Standard guide

- A style guide is a set of standards that outline how code should be written and organized.
- A style guide contains general rules about “how to write” code, e.g. which quotes to use, how many spaces to indent, where to put line breaks, etc. A lot of minor things.
- Style guides are created so new developers can get up to speed on a code base quickly, and then write code that other developers can understand quickly and easily!

Expected style in casing for JavaScript and CSS:

- **CSS:** hyphens
- **JS:** camelCase
- CSS class names should be hyphen separated.
- Either use hexcode or rgb standard for color scheme throughout your application.
- Avoid using px, rem, em at once Choose one. Prefer em.
- Font-weight: 400? What good is it for? Eliminate useless CSS.



Strings in HTML vs JS

- Double quotes in HTML
- Single quotes in JS

When using multiple selectors in a rule declaration, give each selector its own line

Bad

```
.avatar{
    border-radius:50%;
    border:2px solid white; }
.no, .nope, .not_good {
    // ...
}
#lol-no {
  // ...
}
```

Good

```
.avatar {
  border-radius: 50%;
  border: 2px solid white;
}

.one,
.selector,
.per-line {
  // ...
}
```

```
const anakinSkywalker = 'Anakin Skywalker';
const lukeSkywalker = 'Luke Skywalker';

// bad
const obj = {
  episodeOne: 1,
  twoJediWalkIntoACantina: 2,
  lukeSkywalker,
  episodeThree: 3,
  mayTheFourth: 4,
  anakinSkywalker,
};

// good
const obj = {
  lukeSkywalker,
  anakinSkywalker,
  episodeOne: 1,
  twoJediWalkIntoACantina: 2,
  episodeThree: 3,
  mayTheFourth: 4,
};
```

Group your shorthand properties at the beginning of your object declaration.

Makes it simple to read and make decisions. Makes things simpler in a complex nested object structure.

Thank You