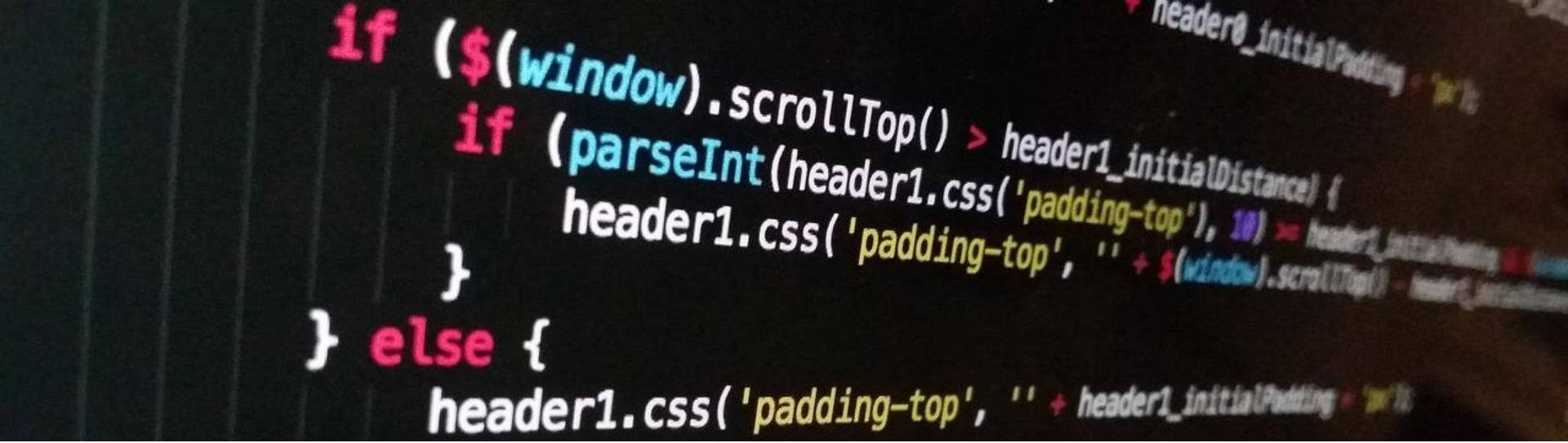

Web Components

Repaso

UNIVERSIDAD ICESI

¿Qué es un Componente?

- *Son un fragmento de la interfaz que **cumple una única función.***
- ***Son reutilizables (principio DRY - Don't Repeat Yourself).***
- ***Son independientes**, tanto de su contexto como del resto de componentes.*
- ***Son autocontenidos**, no filtran estilos o funcionalidad a otros componentes.*
- *Por estas razones, son **más fáciles de mantener y escalar.***

¿Qué son los Componentes Web?

Son un conjunto de tecnologías web que mezcladas permiten realizar bloques de código reutilizable (componentes) que dan como resultado ***elementos personalizados en HTML****.*

Para hacer un Componente Web se utiliza:

1. ***Un HTML Template <template>****, que es una etiqueta que tiene un markup y no se renderiza en la página. Sirve sólo como referencia.*
2. *Un* ***Custom Element****, que es una clase en Javascript, utiliza el* ***HTML Template*** *para ser renderizado y re utilizado. Esta clase provee funciones para el ciclo de vida del componente*
3. *Un* ***Shadow DOM****, que añade a la clase una especie de sombra con HTML encapsulado, renderizado aisladamente del documento principal DOM y controlando su funcionalidad asociada*

La implementación de un Componente Web luce básicamente así:

1. Se crea un elemento **<template>**, que en su HTML interno tendrá el markup y el estilo que estará aislado. Este template será la referencia del markup que renderizará el **Custom Element**.
2. Se crea una clase donde se especifica la funcionalidad que va a tener el **Componente Web** usando la sintaxis de clase.
3. Se registra en nuevo componente personalizado con el método **customElements.define()** que recibe como **primer argumento el nombre que tendrá el tag de este elemento(debe tener un guion "-")** y como **segundo argumento la clase que especifica su funcionalidad**(paso 2).
4. En la clase del **Custom Element**, se añade la sombra usando el método **attachShadow() sobre el elemento de la clase(this)**. A esta sombra **shadowRoot ya disponible como propiedad del elemento** en la clase se agregan los nodos DOM que se renderizarán y harán parte del **Componente Web**. Generalmente estos nodos son un clon del árbol DOM de un elemento **<template>** de referencia(paso 1).
5. Listo! El componente Web personalizado ya puede ser usado en nuestro documento HTML **como cualquier elemento HTML regular.**

Lecturas relacionadas importantes

1. ***Web components MDN(documentación)***

   https://developer.mozilla.org/en-US/docs/Web/API/Web_components

2. ***Ciclos de vida Web Components(tutorial)***

   https://github.com/mappmechanic/super-button-web-component/blob/master/Steps/tutorial2-Lifecycle%20Methods.md

3. ***Añadir interactividad a Web Components(tutorial)***

   https://gomakethings.com/how-to-add-interactivity-to-browser-native-web-components-with-vanilla-js/