Formatters

DEV.F::
DESARROLLAMOS(PERSONAS);

Formatters

Son herramientas de formateo de código que ayudan a mantener la consistencia y legibilidad del código fuente en proyectos de desarrollo de software. Su objetivo principal es formatear automáticamente el código según un conjunto predefinido de reglas de estilo, eliminando así la necesidad de debates sobre convenciones de codificación y ahorrando tiempo en tareas de formateo manual.

DEV.F

```
const formElement = document.querySelector('#language-form');
const ulElement = document.querySelector('#list-languages');
let languagesArray = [];

const STATUS_TYPES = {
    completed: 'finished',
    standBy: 'standBy', start: 'start',
};

formElement.addEventListener("submit", function (event) {

    event.preventDefault();
        const inputElement = event.target.languageElement;
        const radioNodeList =
            event.target.statusRadioElement;
});

const renderViewLanguages = (arr, number,age )=>{
    // do something
};

const getCompleted = (arrLanguages) => {
    const completedList = arrLanguages.filter(element => element.status === STATUS_TYPES.finished);
    const countFinished = completedList.length;
    return countFinished;
};

```

```
const formElement = document.querySelector('#language-form');
const ulElement = document.querySelector('#list-languages');
let languagesArray = [];

const STATUS_TYPES = {
    completed: 'finished',
    standBy: 'standBy',
    start: 'start',
};

formElement.addEventListener('submit', function (event) {

    event.preventDefault();

    // 2. Obtener los valores del formulario
    const inputElement = event.target.languageElement;
    const radioNodeList = event.target.statusRadioElement;


});

const renderViewLanguages = (arrLanguages) => {
    // do something
};

const getCompleted = (arrLanguages) => {
    const completedList = arrLanguages.filter(element => element.status === STATUS_TYPES.finished);
    const countFinished = completedList.length;
    return countFinished;
};

```

Ventajas

- Consistencia del código.
- Ahorro de tiempo.
- Reducción de errores.
- Fácil integración.
- Personalización de reglas.
- Soporte multiplataforma.

Formaters para VSCode

- **ESLint:** Es uno de los Linters más populares para JavaScript. Ayuda a identificar y corregir problemas de estilo, errores de sintaxis y posibles bugs en el código JavaScript.
- **Pylint:** Es un Linter para Python que identifica errores de sintaxis, problemas de estilo y posibles problemas de programación en código Python. Es altamente configurable y puede ayudar a mejorar la calidad del código Python.
- **Stylelint:** Es un Linter para CSS y preprocesadores como Sass y Less. Ayuda a identificar problemas de estilo y errores en código CSS, como reglas no utilizadas, propiedades duplicadas, entre otros.

Linters

DEV.F::
DESARROLLAMOS(PERSONAS);

Linters

Son herramientas de análisis estático de código que se utilizan para identificar problemas de calidad, errores de sintaxis, convenciones de codificación inconsistentes y posibles problemas de seguridad en el código fuente de un programa o proyecto de desarrollo de software.

- Los Linters examinan el código fuente sin ejecutarlo, analizando su estructura y aplicando reglas predefinidas o configurables para identificar posibles problemas.
- Estas reglas pueden abarcar una amplia variedad de aspectos del código, como convenciones de nomenclatura, estilo de código, uso de funciones o métodos específicos, manejo de errores, entre otros.

Ventajas de los linters

Son fáciles de configurar y nos ayudan a detectar problemas como:

- Variables no utilizadas o indefinidas.
- Errores de sintaxis.
- Posibles bugs.
- Identificación de vulnerabilidades de seguridad.
- Cumplir con las convenciones de codificación.

Los Linters se utilizan típicamente como parte del proceso de desarrollo de software para ayudar a mantener la calidad del código y prevenir la introducción de errores y problemas de seguridad. Pueden integrarse con editores de código para proporcionar retroalimentación instantánea a los desarrolladores mientras escriben código

Linters para VSCode

- **ESLint:** Es uno de los Linters más populares para JavaScript. Ayuda a identificar y corregir problemas de estilo, errores de sintaxis y posibles bugs en el código JavaScript.
- **Pylint:** Es un Linter para Python que identifica errores de sintaxis, problemas de estilo y posibles problemas de programación en código Python. Es altamente configurable y puede ayudar a mejorar la calidad del código Python. La extensión de Pylint para VSCode se puede instalar desde el marketplace.
- **Stylelint:** Es un Linter para CSS y preprocesadores como Sass y Less. Ayuda a identificar problemas de estilo y errores en código CSS, como reglas no utilizadas, propiedades duplicadas, entre otros. La extensión Stylelint para VSCode se puede instalar desde el marketplace.

Notas para configuración

DEV.F::
DESARROLLAMOS(PERSONAS);

Notas para configuración

- **Configuración base de VSCode:** Se guarda en el **settings.json** y **editor.config** de vscode y nos permite definir el estándar de codificación de nuestro editor.
- **Eslint:** Es una herramienta de resaltado de sintaxis y es customizable con reglas que se pueden definir en el archivo **.eslintrc.json**.
- **Prettier:** Es un formateador de código que puede autocorregir automaticamente nuestro código según el conjunto de reglas que establezcamos en el archivo **.prettierrc.json.**

¿En qué momento se pueden usar?

- Al guardar.
- Al pegar código.
- Al ejecutar un comando npm (programado por nosotros).
- Antes de hacer un commit.
- Antes de desplegar a algún entorno.

Docs

- Prettier.
- Como configurar Prettier en VSCode.
- ESLint.
- Como configurar ESLint en VSCode.