**Test de Evaluación de Conocimientos en SASS**

**Parte 1: Preguntas de opción múltiple (Teoría)**

1. **¿Cuál es la diferencia principal entre SASS y SCSS?**
   * a) SASS usa llaves {} y SCSS no.
   * b) SASS es la versión antigua y SCSS es una nueva sintaxis que utiliza llaves {} y puntos y comas.
   * c) SCSS no permite anidación.
   * d) SCSS no soporta variables.

RESPUESTA: \_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. **¿Cuál es el propósito de un mixin en SASS?**
   * a) Reutilizar bloques de código con pequeñas modificaciones.
   * b) Definir variables globales.
   * c) Agrupar selectores CSS.
   * d) Aplicar estilos condicionales.

RESPUESTA: \_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. **¿Qué hace la función @extend en SASS?**
   * a) Asigna el valor de una variable a otro selector.
   * b) Combina las propiedades de un selector en otro.
   * c) Reemplaza variables en el código.
   * d) Crea un nuevo selector.

RESPUESTA: \_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. **¿Cómo puedes acceder al valor de una clave en un map de SASS?**
   * a) @get(map, clave);
   * b) map-access(map, clave);
   * c) map-get(map, clave);
   * d) map-value(map, clave);

RESPUESTA: \_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. **¿Qué hace la directiva @import en SASS?**
   * a) Importa variables desde otro archivo CSS.
   * b) Combina varios archivos CSS en uno solo.
   * c) Permite la importación de archivos SASS o SCSS en otros archivos SASS.
   * d) Es equivalente a la regla CSS @import.

RESPUESTA: \_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Parte 2: Verdadero o Falso**

1. **SASS permite la creación de funciones personalizadas para realizar cálculos complejos en hojas de estilo.**
   * Verdadero
   * Falso

RESPUESTA: \_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. **La anidación en SASS se utiliza únicamente para anidar selectores, no propiedades.**
   * Verdadero
   * Falso

RESPUESTA: \_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. **El operador & en SASS hace referencia al selector padre.**

* Verdadero
* Falso

RESPUESTA: \_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. **SASS compila directamente a JavaScript.**

* Verdadero
* Falso

RESPUESTA: \_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. **Puedes usar la directiva @media para anidar consultas de medios (media queries) dentro de otros selectores en SASS.**

* Verdadero
* Falso

RESPUESTA: \_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Parte 3: Preguntas prácticas (Copiar y pegar la Respuesta)**

1. **Variables en SASS**: Crea una variable que almacene el color #ff6347 (tomato) y úsala en un selector de clase llamado .button.

RESPUESTA:(Código):

1. **Mixin básico**: Crea un mixin llamado border-radius que acepte un parámetro llamado $radius, y aplícalo a un selector .box.

RESPUESTA:(Código):

1. **Map y acceso a claves**: Define un map que contenga tres colores: primary con el valor #3498db, secondary con el valor #2ecc71 y danger con el valor #e74c3c. Luego, escribe una regla CSS que aplique el color primary de ese mapa a una clase .alert.

RESPUESTA:(Código):

1. **@extend y reutilización de estilos**: Usa @extend para compartir los estilos de un selector .card con un selector .profile-card.

RESPUESTA:(Código):

1. **Media queries con SASS**: Escribe una consulta de medios (media query) en SASS que aplique un ancho de 100% a un contenedor .container en pantallas menores a 768px.

RESPUESTA:(Código):

**Parte 4: Resolución de problemas**

1. **Error en código SASS**: Observa el siguiente código SASS y corrige el error.

*@mixin text-style($color, $size)*

*color: $color;*

*font-size: $size;*

*.button {*

*@include text-style(blue, 16px);*

*padding 10px;*

*background-color: white;*

*}*

RESPUESTA:(Código):

1. **Optimización con mixins**: Optimiza el siguiente código usando un mixin para evitar la duplicación de estilos.

*.button-primary {*

*background-color: blue;*

*color: white;*

*padding: 10px;*

*}*

*.button-secondary {*

*background-color: gray;*

*color: white;*

*padding: 10px;*

*}*

RESPUESTA:(Código):

**Guía de Calificación**

**Parte 1: Opción Múltiple (5 preguntas, 1 punto cada una)**

* Cada respuesta correcta suma 1 punto.

**Parte 2: Verdadero o Falso (5 preguntas, 1 punto cada una)**

* Cada respuesta correcta suma 1 punto.

**Parte 3: Preguntas prácticas (5 preguntas, 2 puntos cada una)**

* La respuesta correcta completa suma 2 puntos. Si hay algún error menor, puedes dar 1 punto parcial.

**Parte 4: Resolución de problemas (2 preguntas, 3 puntos cada una)**

* Cada respuesta correcta suma 3 puntos. Si hay algún error menor o no se resuelve por completo, puedes dar puntuaciones parciales (2 o 1 punto).

**Puntuación**

* **0-15 puntos**: Nivel básico. Requiere más práctica en conceptos fundamentales.
* **16-25 puntos**: Nivel intermedio. Tienes buena comprensión, pero puedes mejorar en algunas áreas.
* **26-30 puntos**: Nivel avanzado. Excelente manejo de SASS y buenas prácticas en su uso.