CSS- 09- g- Conditions

SASS permet d'inclure des conditions @if, @else et @else if similaire aux conditions en JavaScript. Cette fonctionnalité permet d'écrire du code plus complexe plus facilement.

@if

La règle @if permet de contrôler l'exécution d'un bloc de style en fonction d'une condition.

Par exemple, une mixin qui permettrait de générer un avatar carré ou circulaire en fonction de la valeur du paramètre rounded *(true ou false)*:

```
@mixin avatar($size: 100px, $rounded: false) {
  width: $size;
  height: $size;
  background-size: cover;
  background-position: 50% 50%;

  @if $rounded {
    border-radius: 50%;
  }
}
```

Pour obtenir un avatar carré de 100px, il serait possible d'écrire:

```
.avatar-facebook {
  @include avatar; // Aucun paramètre. Tout par défaut
  background-image: url('facebook-logo.png');
}
```

Tandis que pour obtenir un avatar rond de 200px:

```
.avatar-twitter {
  @include avatar(200px, true);
  background-image: url('twitter-logo.png');
}
```

Ce qui produirait les codes suivant:

```
.avatar-facebook {
  width: 100px;
  height: 100px;
  background-size: cover;
  background-position: 50% 50%;
  background-image: url('facebook-logo.png');
}

.avatar-twitter {
  width: 200px;
  height: 200px;
  background-size: cover;
  background-position: 50% 50%;
  border-radius: 50%;
  background-image: url('twitter-logo.png');
}
```

- @if

@else

La règle @else permet de déclencher l'exécution d'un bloc de style lorsqu'une condition @if n'est pas respectée.

Par exemple, une mixin de choix de thème de couleur:

```
@mixin dark-theme($dark: true) {
  @if $dark {
    background: black;
    color: white;
  } @else {
    background: white;
    color: black;
  }
}
```

Pour obtenir un menu blanc ◯ avec texte noir ⬤ , il serait possible d'écrire:

```
.menu {
```

```
  @include dark-theme(false);
}
```

Ce qui générerait le code suivant:

```
.menu {
  background: white;
  color: black;
}
```

- @else

@else if

Par exemple, si pour créer une mixin permettant de gérer les breakpoints CSS d'un site à partir de mots-clés plutôt que de chiffres qui sont parfois difficiles à retenir, il serait possible de faire:

```
@mixin breakpoint($size: sm) {
  @if $size == sm {
    @media (min-width: 576px) { @content; }
  } @else if $size == md {
    @media (min-width: 768px) { @content; }
  } @else if $size == lg {
    @media (min-width: 992px) { @content; }
  } @else if $size == xl {
    @media (min-width: 1200px) { @content; }
  }
}
```

Remarquez @content qui permet de récupérer tout ce qui se trouve à l'intérieur du @include de la mixin.

Par exemple:

```
@include breakpoint(md) {
  body { font-size: 18px; }
}
```

Générera le code suivant:

```
@media (min-width: 768px) {
  body { font-size: 18px; }
}
```

Cependant, vous n'avez pas à retenir le chiffre 768px. Mieux, si votre équipe décide de changer la valeur de *md* à 780px, elle peut simplement la changer dans la mixin et la valeur ce mettra à jours partout sans que personne n'aille à appliquer de modification à son code.

- @else-if