CSS - AURAL MEDIA

A web document can be rendered by a speech synthesizer. CSS2 allows you to attach specific sound style features to specific document elements.

Aural rendering of documents is mainly used by the visually impaired. Some of the situations in which a document can be accessed by means of aural rendering rather than visual rendering are the following.

- Learning to read
- Training
- Web access in vehicles
- Home entertainment
- Industrial documentation
- Medical documentation

When using aural properties, the canvas consists of a three-dimensional physical space *soundsurrounds* and a temporal space *onemayspecifysoundsbefore, during, andafterothersounds*.

The CSS properties also allow you to vary the quality of synthesized speech *voicetype, frequency, inflection, etc.*

Here is an example −

```
<html>
   <head>

      <style tyle="text/css">
         h1, h2, h3, h4, h5, h6 {
            voice-family: paul;
            stress: 20;
            richness: 90;
            cue-before: url("../audio/pop.au");
         }
         p{
            azimuth:center-right;
         }
      </style>

   </head>
   <body>

      <h1>Tutorialspoint.com</h1>
      <h2>Tutorialspoint.com</h2>
      <h3>Tutorialspoint.com</h3>
      <h4>Tutorialspoint.com</h4>
      <h5>Tutorialspoint.com</h5>
      <h6>Tutorialspoint.com</h6>
      <p>Tutorialspoint.com</p>

   </body>
</html>
```

It will produce the following result −

It will direct the speech synthesizer to speak headers in a voice (a kind of *audio font*) called "paul", on a flat tone, but in a very rich voice. Before speaking the headers, a sound sample will be played from the given URL.

Now, we will see various properties related to aural media.

- The **azimuth** property sets, where the sound should come from horizontally.
- The **elevation** property sets, where the sound should come from vertically.
- The **cue-after** specifies a sound to be played after speaking an element's content to delimit it from other.
- The **cue-before** specifies a sound to be played before speaking an element's content to delimit it from other.
- The **cue** is a shorthand for setting cue-before and cue-after.
- The **pause-after** specifies a pause to be observed after speaking an element's content.
- The **pause-before** specifies a pause to be observed before speaking an element's content.
- The **pause** is a shorthand for setting pause-before and pause-after.
- The **pitch** specifies the average pitch *afrequency* of the speaking voice.
- The **pitch-range** specifies variation in average pitch.
- The **play-during** specifies a sound to be played as a background while an element's content is spoken.
- The **richness** specifies the richness, or brightness, of the speaking voice.
- The **speak** specifies whether text will be rendered aurally and if so, in what manner.
- The **speak-numeral** controls how numerals are spoken.
- The **speak-punctuation** specifies how punctuation is spoken.
- The **speech-rate** specifies the speaking rate.
- The **stress** specifies the height of "local peaks" in the intonation contour of a voice.
- The **voice-family** specifies the prioritized list of voice family names.
- The **volume** refers to the median volume of the voice.

The azimuth Property

The azimuth property sets where the sound should come from horizontally. The possible values are listed below −

- **angle** − Position is described in terms of an angle within the range *-360deg* to *360deg*. The value *0deg* means directly ahead in the center of the sound stage. *90deg* is to the right, *180deg* behind, and *270deg* (or, equivalently and more conveniently, *-90deg*) to the left.
- **left-side** − Same as '270deg'. With 'behind', '270deg'.

- **far-left** − Same as '300deg'. With 'behind', '240deg'.
- **left** − Same as '320deg'. With 'behind', '220deg'.
- **center-left** − Same as '340deg'. With 'behind', '200deg'.
- **center** − Same as '0deg'. With 'behind', '180deg'.
- **center-right** − Same as '20deg'. With 'behind', '160deg'.
- **right** − Same as '40deg'. With 'behind', '140deg'.
- **far-right** − Same as '60deg'. With 'behind', '120deg'.
- **right-side** − Same as '90deg'. With 'behind', '90deg'.
- **leftwards** − Moves the sound to the left and relative to the current angle. More precisely, subtracts 20 degrees.
- **rightwards** − Moves the sound to the right, relative to the current angle. More precisely, adds 20 degrees.

Here is an example −

```
<style tyle="text/css">
   <!--
   h1   { azimuth: 30deg }
   td.a { azimuth: far-right }          /*  60deg */
   #12  { azimuth: behind far-right }   /* 120deg */
   p.comment { azimuth: behind }        /* 180deg */
   -->
</style>
```

The elevation Property

The elevation property sets where the sound should come from vertically. The possible values are as follows −

- **angle** − Specifies the elevation as an angle, between *-90deg* and *90deg*. *0deg* means on the forward horizon, which loosely means level with the listener. *90deg* means directly overhead and *-90deg* means directly below.
- **below** − Same as '-90deg'.
- **level** − Same as '0deg'.
- **above** − Same as '90deg'.
- **higher** − Adds 10 degrees to the current elevation.
- **lower** − Subtracts 10 degrees from the current elevation.

Here is an example −

```
<style tyle="text/css">
   <!--
   h1   { elevation: above }
   tr.a { elevation: 60deg }
   tr.b { elevation: 30deg }
   tr.c { elevation: level }
   -->
</style>
```

The cue-after Property

The cue-after property specifies a sound to be played after speaking an element's content to delimit it from other. The possible values include −

- **url** − The URL of a sound file to be played.
- **none** − Nothing has to be played.

Here is an example −

```
<style tyle="text/css">
   <!--
   a {cue-after: url("dong.wav");}
   h1 {cue-after: url("pop.au"); }
   -->
</style>
```

The cue-before Property

This property specifies a sound to be played before speaking an element's content to delimit it from other. The possible values are −

- **url** − The URL of a sound file to be played.
- **none** − Nothing has to be played.

Here is an example −

```
<style tyle="text/css">
   <!--
   a {cue-before: url("bell.aiff");}
   h1 {cue-before: url("pop.au"); }
   -->
</style>
```

The cue Property

The cue property is a shorthand for setting *cue-before* and *cue-after*. If two values are given, the first value is *cue-before* and the second is *cue-after*. If only one value is given, it applies to both properties.

For example, the following two rules are equivalent −

```
<style tyle="text/css">
   <!--
   h1 {cue-before: url("pop.au"); cue-after: url("pop.au") }
   h1 {cue: url("pop.au") }
   -->
</style>
```

The pause-after Property

This property specifies a pause to be observed after speaking an element's content. The possible values are −

- **time** − Expresses the pause in absolute time units *secondsandmilliseconds*.
- **percentage** − Refers to the inverse of the value of the *speech-rate* property. For example, if the speech-rate is 120 words per minute *i. e. awordtakeshalfasecond*, *or*500*ms*, then a *pause-after* of 100% means a pause of 500 ms and a *pause-after* of 20% means 100ms.

The pause-before Property

This property specifies a pause to be observed before speaking an element's content. The possible values are −

- **time** − Expresses the pause in absolute time units *secondsandmilliseconds*.
- **percentage** − Refers to the inverse of the value of the *speech-rate* property. For example, if

the speech-rate is 120 words per minute *i. e. awordtakeshalfasecond*, *or*500*ms*, then a *pause-before* of 100% means a pause of 500 ms and a *pause-before* of 20% means 100ms.

The pause Property

This property is a shorthand for setting *pause-before* and *pause-after*. If two values are given, the first value is *pause-before* and the second is pause-after.

Here is an example −

```
<style tyle="text/css">
   <!--
   /* pause-before: 20ms; pause-after: 20ms */
   h1 { pause : 20ms }

   /* pause-before: 30ms; pause-after: 40ms */
   h2{ pause : 30ms 40ms }

   /* pause-before: ?; pause-after: 10ms */
   h3 { pause-after : 10ms }
   -->
</style>
```

The pitch Property

This property specifies the average pitch *afrequency* of the speaking voice. The average pitch of a voice depends on the voice family. For example, the average pitch for a standard male voice is around 120Hz, but for a female voice, it's around 210Hz. The possible values are −

- **frequency** - Specifies the average pitch of the speaking voice in hertz *Hz*.
- **x-low, low, medium, high, x-high** - These values do not map to absolute frequencies since these values depend on the voice family.

The pitch-range Property

This property specifies variation in average pitch. The possible values are −

- **number** − A value between '0' and '100'. A pitch range of '0' produces a flat, monotonic voice. A pitch range of 50 produces normal inflection. Pitch ranges greater than 50 produce animated voices.

The play-during Property

This property specifies a sound to be played as a background while an element's content is spoken. Possible values could be any of the followings −

- **URI** − The sound designated by this <uri> is played as a background while the element's content is spoken.
- **mix** − When present, this keyword means that the sound inherited from the parent element's *play-during* property continues to play and the sound designated by the *uri* is mixed with it. If *mix* is not specified, the element's background sound replaces the parent's.
- **repeat** − When present, this keyword means that the sound will repeat if it is too short to fill the entire duration of the element. Otherwise, the sound plays once and then stops.
- **auto** − The sound of the parent element continues to play.
- **none** − This keyword means that there is silence.

Here is an example −

```
<style tyle="text/css">
   <!--
   blockquote.sad { play-during: url("violins.aiff") }
```

```
   blockquote q   { play-during: url("harp.wav") mix }
   span.quiet     { play-during: none }
   -->
</style>
```

The richness Property

This property specifies the richness or brightness of the speaking voice. The possible values are:

- **number** − A value between '0' and '100'. The higher the value, the more the voice will carry. A lower value will produce a soft, mellifluous voice.

The speak Property

This property specifies whether text will be rendered aurally and if so, in what manner. The possible values are −

- **none** − Suppresses aural rendering so that the element requires no time to render.
- **normal** − Uses language-dependent pronunciation rules for rendering an element and its children.
- **spell-out** − Spells the text one letter at a time.

Note the difference between an element whose 'volume' property has a value of 'silent' and an element whose 'speak' property has the value 'none'. The former takes up the same time as if it had been spoken, including any pause before and after the element, but no sound is generated. The latter requires no time and is not rendered.

The speak-numeral Property

This property controls how numerals are spoken. The possible values are −

- **digits** − Speak the numeral as individual digits. Thus, "237" is spoken "Two Three Seven".
- **continuous** − Speak the numeral as a full number. Thus, "237" is spoken "Two hundred thirty seven". Word representations are language-dependent.

The speak-punctuation Property

This property specifies how punctuation is spoken. The possible values are −

- **code** − Punctuation such as semicolons, braces, and so on are to be spoken literally.
- **none** − Punctuation is not to be spoken, but instead rendered naturally as various pauses.

The speech-rate property

This property specifies the speaking rate. Note that both absolute and relative keyword values are allowed. The possible values are −

- **number** − Specifies the speaking rate in words per minute.
- **x-slow** − Same as 80 words per minute.
- **slow** − Same as 120 words per minute.
- **medium** − Same as 180 - 200 words per minute.
- **fast** − Same as 300 words per minute.
- **x-fast** − Same as 500 words per minute.
- **faster** − Adds 40 words per minute to the current speech rate.
- **slower** − Subtracts 40 words per minutes from the current speech rate.

The stress Property

This property specifies the height of "local peaks" in the intonation contour of a voice. English is a stressed language, and different parts of a sentence are assigned primary, secondary, or tertiary stress. The possible values are −

- **number** − A value between '0' and '100'. The meaning of values depends on the language being spoken. For example, a level of '50' for a standard, English-speaking male voice $averagepitch = 122Hz$, speaking with normal intonation and emphasis would have a different meaning than '50' for an Italian voice.

The voice-family Property

The value is a comma-separated, prioritized list of voice family names. It can have following values:

- **generic-voice** − Values are voice families. Possible values are 'male', 'female', and 'child'.
- **specific-voice** − Values are specific instances $e.g.$, *comedian, trinoids, carlos, lani*.

Here is an example −

```
<style tyle="text/css">
   <!--
   h1 { voice-family: announcer, male }
   p.part.romeo  { voice-family: romeo, male }
   p.part.juliet { voice-family: juliet, female }
   -->
</style>
```

The volume Property

Volume refers to the median volume of the voice. It can have following values −

- **numbers** − Any number between '0' and '100'. '0' represents the minimum audible volume level and 100 corresponds to the maximum comfortable level.
- **percentage** − These values are calculated relative to the inherited value, and are then clipped to the range '0' to '100'.
- **silent** − No sound at all. The value '0' does not mean the same as 'silent'.
- **x-soft** − Same as '0'.
- **soft** − Same as '25'.
- **medium** − Same as '50'.
- **loud** − Same as '75'.
- **x-loud** − Same as '100'.

Here is an example −

```
<style tyle="text/css">
   <!--
   P.goat  { volume: x-soft }
   -->
</style>
```

Paragraphs with class **goat** will be very soft.