

Web Audio Codelab #2

Data Structure, Function, and Iteration

Music 220A Summer 2021

Google



array javascript mdn



Google Search

I'm Feeling Lucky

```
/* Array in JavaScript */  
  
const array = [1, 2, 3, 4];  
  
console.log(array[0]);    // prints "1"  
  
array.push(5);           // makes [1, 2, 3, 4, 5]  
  
console.log(array.length); // prints "5"
```

```
/* Object in JavaScript */
```

```
const partial1 = {  
  frequency: 440,  
  amplitude: 1.0  
};
```

```
console.log(partial1.frequency); // prints "440"  
console.log(partial1.amplitude); // prints "1.0"
```

```
// overwrites the value with "220"  
partial1.frequency = 220;
```

```
/* Function in JavaScript */
```

```
function doStuff(args) { /* code */ }
```

```
const doStuff = function (args) { /* code */ }
```

```
const doStuff = (args) => { /* code */ }
```

```
function () { /* code */ }
```

```
(function () { /* code */ })()
```

```
/* Callback function */  
  
function doStuff() { /* code */ }  
  
const buttonElement = document.querySelector('#start-audio');  
buttonElement.addEventListener('click', doStuff);  
  
window.addEventListener('load', () => { /* code */ });
```

```
/* Iteration in JavaScript (1) */
```

```
const freqs = [220, 440, 660, 880];
```

```
const amps = [1.0, 0.5, 0.33, 0.25];
```

```
for (let i = 0; i < freqs.length; ++i) {
```

```
    console.log(freqs[i], amps[i]);
```

```
}
```

```
/* Iteration in JavaScript (2) */
```

```
const freqs = [220, 440, 660, 880];
```

```
const amps = [1.0, 0.5, 0.33, 0.25];
```

```
freqs.forEach((value, index) => {  
    console.log(value, amps[index]);  
});
```



```
/* Iteration in JavaScript (3) */
```

```
const tone = {  
  freq1: 220, amp1: 1.0,  
  freq2: 440, amp2: 0.5,  
  freq3: 660, amp3: 0.33,  
  freq4: 880, amp4: 0.25,  
};  
  
for (const key in tone) {  
  console.log(key, tone[key]);  
}
```

```
/* Iteration in JavaScript (4) */
```

```
const tone = [  
  {freq: 220, amp: 1.0},  
  {freq: 440, amp: 0.5},  
  {freq: 660, amp: 0.33},  
  {freq: 880, amp: 0.25},  
];  
  
tone.forEach((partial, index) => {  
  console.log(index, partial.freq, partial.amp);  
});
```

Tutorial:

AudioParam

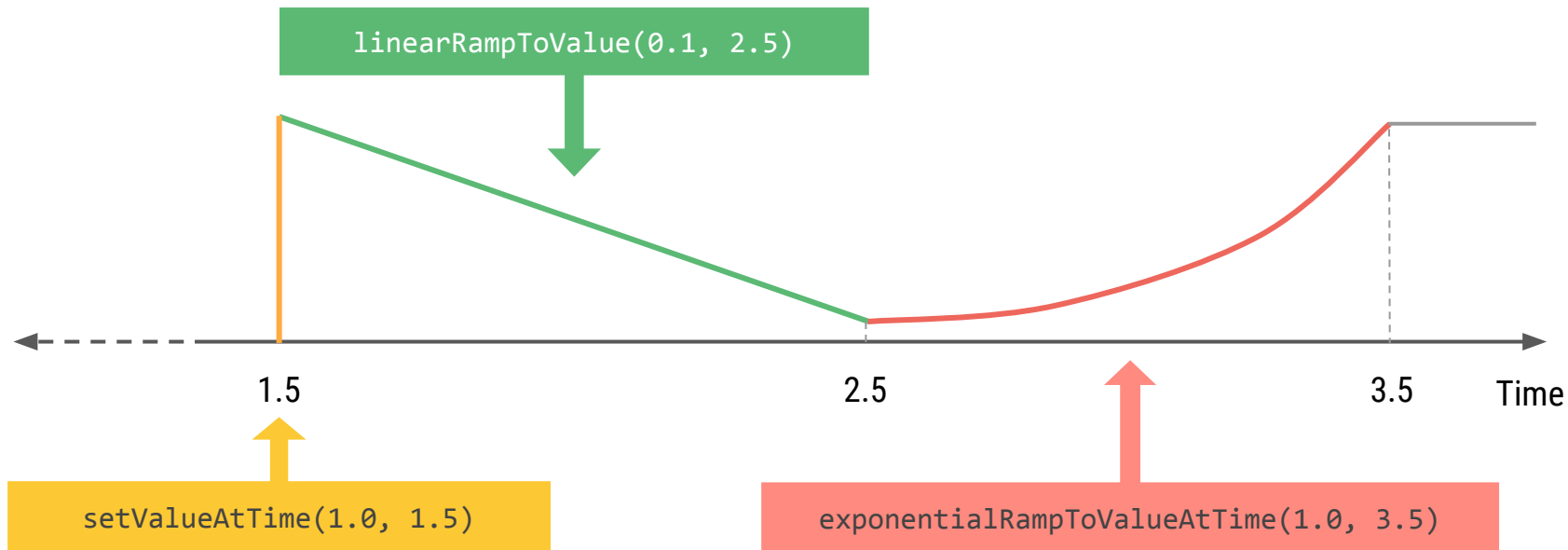
AudioParam

- Member of AudioNode
- Automation methods
 - `AudioParam.setValueAtTime(value, startTime)`
 - `AudioParam.linearRampToValueAtTime(value, endTime)`
 - `AudioParam.exponentialRampToValueAtTime(value, endTime)`
 - `AudioParam.setTargetAtTime(value, startTime, timeConstant)`

```
const amp = new GainNode(context);

amp.gain.setValueAtTime(1.0, 1.5);
amp.gain.linearRampToValue(0.1, 2.5);
amp.gain.exponentialRampToValueAtTime(1.0, 3.5);
```

AudioParam



Examples:

Additive Synthesis

ccrma.github.io/music220a/03-additive