

Hello,

KDT 웹 개발자 양성 프로젝트

5기!

with





LocalStorage & SessionStorage



LocalStorage 에 값 저장하기!

- `window.localStorage.setItem('key', 'value')` 로 값을 저장 가능

```
<script>  
  window.localStorage.setItem('name', '이효석');  
</script>
```



LocalStorage 의 값 불러오기

- `window.localStorage.getItem('key')` 로 값을 저장 가능합니다!

```
<script>  
  window.localStorage.setItem('name', '이효석');  
  console.log(window.localStorage.getItem('name'));  
</script>
```

이효석





```
<script>
  const tetz = {
    name: "이효석",
    isOld: true,
    hobby: ["음주", "콜라", "플스", "자전거"],
  };

  const jsonObj = JSON.stringify(tetz);
  window.localStorage.setItem('tetz', jsonObj);

  const obj = JSON.parse(window.localStorage.getItem('tetz'));
  console.log(obj);
</script>
```

```
▼ {name: '이효석', isOld: true, hobby: Array(4)} ⓘ
  ▶ hobby: (4) ['음주', '콜라', '플스', '자전거']
    isOld: true
    name: "이효석"
  ▶ [[Prototype]]: Object
```



LocalStorage 에 특정 키 삭제하기!

```
<script>
  window.localStorage.setItem('tetz', '테츠에요!');
  window.localStorage.setItem('pororo', '뽀로로에요!');
  window.localStorage.removeItem('pororo');
  console.log(window.localStorage);
</script>
```

```
▼ Storage {tetz: '테츠에요!', length: 1} ⓘ
  tetz: "테츠에요!"
  length: 1
  ▶ [[Prototype]]: Storage
```



LocalStorage 전체 삭제하기!

```
<script>
  window.localStorage.setItem('tetz', '테츠에요!');
  window.localStorage.setItem('pororo', '뽀로로에요!');
  window.localStorage.clear();
  console.log(window.localStorage);
</script>
```

```
▼ Storage {Length: 0} ⓘ
  length: 0
  ► [[Prototype]]: Storage
```

LocalStorage 에 저장 된 데이터 개수 구하기



```
<script>
  window.localStorage.setItem('tetz', '테츠에요!');
  window.localStorage.setItem('pororo', '뽀로로에요!');
  console.log(window.localStorage.length);
</script>
```

2



LocalStorage 에 저장 된 키 이름 찾기!

```
<script>
  window.localStorage.setItem('tetz', '테츠에요!');
  window.localStorage.setItem('pororo', '뽀로로에요!');
  console.log(window.localStorage.key(0));
  console.log(window.localStorage.key(1));
</script>
```

tetz

pororo



New in v5.3

Color mode support, expanded color palette, and more!



Build fast, responsive sites with Bootstrap

Powerful, extensible, and feature-packed frontend toolkit. Build and customize with Sass, utilize prebuilt grid system and components, and bring projects to life with powerful JavaScript plugins.

```
$ npm i bootstrap@5.3.0-alpha1
```



 [Read the docs](#)

<https://getbootstrap.com/>

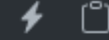


```
<link
  href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css"
  rel="stylesheet"
  integrity="sha384-GLh1TQ8iRABdZL1603oVMWSktQOp6b7In1Zl3/Jr59b6EGGoI1aFkw7cmDA6j6gD"
  crossorigin="anonymous"
/>
<script
  defer
  src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/js/bootstrap.min.js"
  integrity="sha384-mQ93GR66B00ZXjt0Y05KlohRA5SY2XofN4zfuZxLkoj1gXtW8ANNCe9d5Y3eG5eD"
  crossorigin="anonymous"
></script>
```

- Defer 를 쓰면 되므로 <head> 태그 내부에 삽입!



HTML



```
<form>
  <div class="mb-3">
    <label for="exampleInputEmail1" class="form-label">Email address</label>
    <input type="email" class="form-control" id="exampleInputEmail1" aria-describedby="emailHel
    <div id="emailHelp" class="form-text">We'll never share your email with anyone else.</div>
  </div>
  <div class="mb-3">
    <label for="exampleInputPassword1" class="form-label">Password</label>
    <input type="password" class="form-control" id="exampleInputPassword1">
  </div>
  <div class="mb-3 form-check">
    <input type="checkbox" class="form-check-input" id="exampleCheck1">
    <label class="form-check-label" for="exampleCheck1">Check me out</label>
  </div>
  <button type="submit" class="btn btn-primary">Submit</button>
</form>
```





Input group

Floating labels

Layout

Validation

Components

Accordion

Alerts

Badge

Breadcrumb

Buttons

Button group

Card

Carousel

Close button

Collapse

Dropdowns

List group

Modal

Navbar

Examples

Bootstrap includes several predefined button styles, each serving its own semantic purpose, with a few extras thrown in for more control.

Primary

Secondary

Success

Danger

Warning

Info

Light

Dark

[Link](#)

HTML



```
<button type="button" class="btn btn-primary">Primary</button>
<button type="button" class="btn btn-secondary">Secondary</button>
<button type="button" class="btn btn-success">Success</button>
<button type="button" class="btn btn-danger">Danger</button>
<button type="button" class="btn btn-warning">Warning</button>
<button type="button" class="btn btn-info">Info</button>
<button type="button" class="btn btn-light">Light</button>
<button type="button" class="btn btn-dark">Dark</button>

<button type="button" class="btn btn-link">Link</button>
```



클래스로 컨트롤 하기



```
<div class="d-flex flex-column align-items-center parent">  
  <div class="mt-4 p-3 child">1</div>  
  <div class="mt-4 child">2</div>  
  <div class="mt-4 child">3</div>  
  <div class="mt-4 child">4</div>  
</div>
```

1

2

3

4

```
<style>  
  .parent {  
    padding: 50px 50px;  
  }  
  .child {  
    border: 1px solid black;  
    width: 100px;  
  }  
</style>
```



Grid



```
<div class="container" style="background-color: orange">  
  <div class="fixed">fixed width (.container)</div>  
</div>  
<br />  
<div class="container-fluid" style="background-color: skyblue">  
  <div class="fluid">full width (.container-fluid)</div>  
</div>
```

fixed width (.container)

full width (.container-fluid)



Container

- 화면 넓이 1400px 이상 → 넓이 1320px
- 화면 넓이 1400 ~ 1200px → 넓이 1140px
- 화면 넓이 1200 ~ 992px → 넓이 960px

	Extra small <576px	Small ≥576px	Medium ≥768px	Large ≥992px	X-Large ≥1200px	XX-Large ≥1400px
<code>.container</code>	100%	540px	720px	960px	1140px	1320px



Getting started

An overview of Bootstrap, how to download and use, basic templates and examples, and more.

Demystify performance metrics. [Play free!](#)

Processes, Apps, Hosts. Monitor it All via Carbon.

Download

Bootstrap (currently v3.3.4) has a few easy ways to quickly get started, each one appealing to a different skill level and use case. Read through to see what suits your particular needs.

Bootstrap

Compiled and minified CSS, JavaScript, and fonts. No docs or original source files are included.

[Download Bootstrap](#)

Source code

Source Less, JavaScript, and font files, along with our docs. **Requires a Less compiler and some setup.**

[Download source](#)

Sass

Bootstrap ported from Less to Sass for easy inclusion in Rails, Compass, or Sass-only projects.

[Download Sass](#)

Bootstrap CDN

The folks over at [MaxCDN](#) graciously provide CDN support for Bootstrap's CSS and JavaScript. Just use these [Bootstrap CDN](#) links.

```
<!-- Latest compiled and minified CSS -->
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.4/css/bootstrap.min.css">

<!-- Optional theme -->
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.4/css/bootstrap-theme.min.css">

<!-- Latest compiled and minified JavaScript -->
```

[Copy](#)

- Download
- What's included
- Compiling CSS and JavaScript
- Basic template
- Examples
- Tools
- Community
- Creating responsiveness
- Migrating from 2.x to 3.0
- Browser and device support
- Third party support
- Accessibility
- License FAQs
- Translations
- Back to top

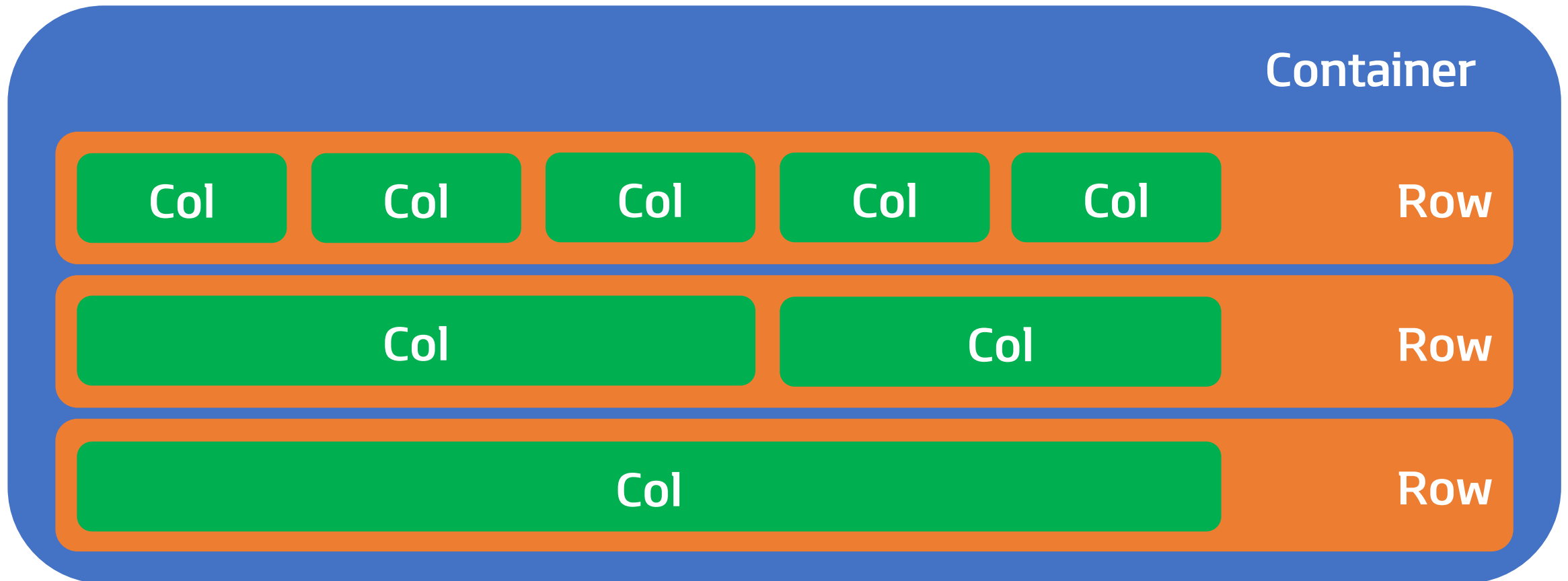
Bootstrap Grid System

그리드 레이아웃을 구성 시에는 반드시 **.row** (행)를 먼저 배치하고 행 안에 **.col-**-*** (열)을 필요한 갯수만큼 배치한다. 즉, container 내에 .row(행)을 먼저 배치하고 그 안에 .col-**-*(열)을 배치한다. 그리고 콘텐츠는 .col-**-* 내에 배치한다.



Col(umn)

- 컨테이너 내부에 위치 한 Row 안에 위치하는 열의 개념





반응형



```
<div class="container">
  <div class="row">
    <div class="col-lg-4 col-md-6 col-sm-12">1</div>
    <div class="col-lg-4 col-md-6 col-sm-12">2</div>
    <div class="col-lg-4 col-md-6 col-sm-12">3</div>
    <div class="col-lg-4 col-md-6 col-sm-12">4</div>
    <div class="col-lg-4 col-md-6 col-sm-12">5</div>
    <div class="col-lg-4 col-md-6 col-sm-12">6</div>
  </div>
</div>
```



지금 시작합니다



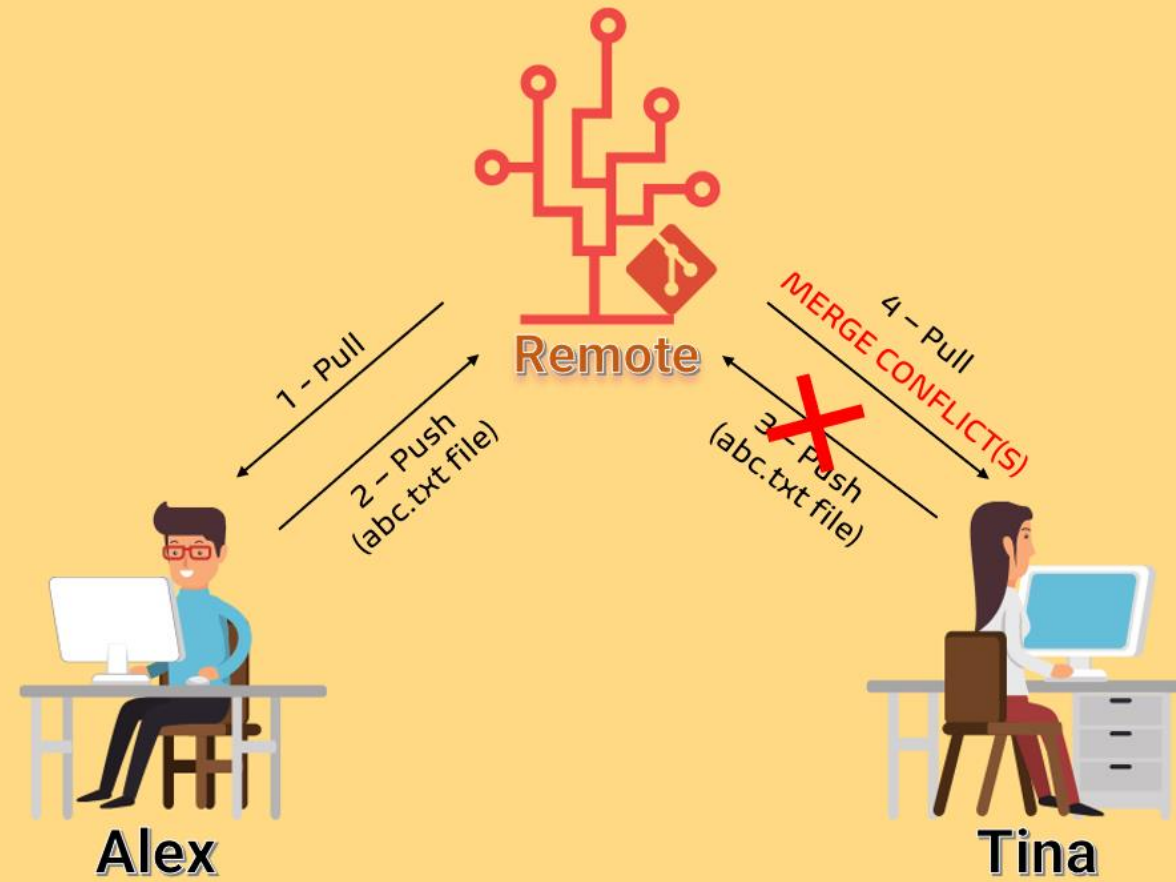
Git Ignore



```
# style.css
```

```
# style.css.map
```

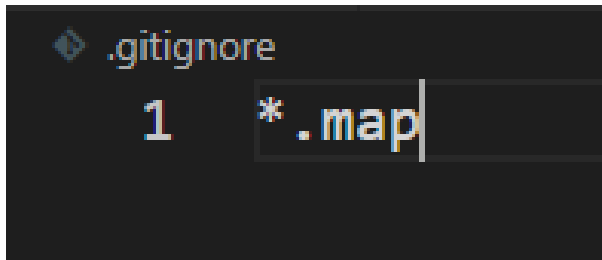
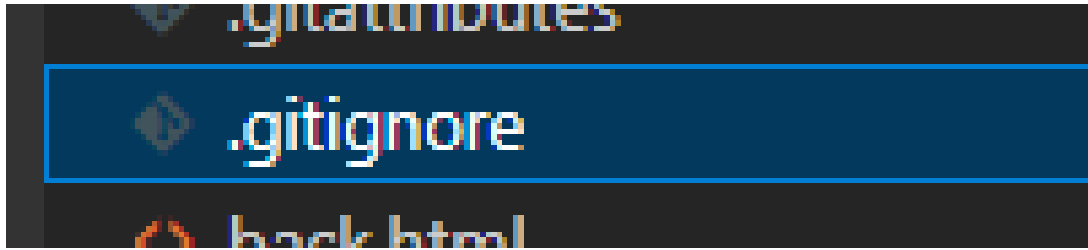
```
{  
  "version": 3,  
  "sources": [  
    "_reset.scss",  
    "style.css",  
    "style.scss",  
    "_mixin.scss"  
  ],  
  "names": [],  
  "mappings": "AACA;EACE,WAAA;EACA,YAAA;EACA,sBAAA;A  
  "file": "style.css"  
}
```





무시





- 파일 하나만 무시하고 싶을 때

```
fileName.txt
```

- 특정 디렉토리의 특정 파일을 무시하고 싶을 때

```
fileDirectory/fileName.txt
```

- 특정 디렉토리의 모든 파일을 무시하고 싶을 때

```
fileDirectory/
```

- 특정 확장자를 가진 모든 파일을 무시하고 싶을 때

```
*.txt
```

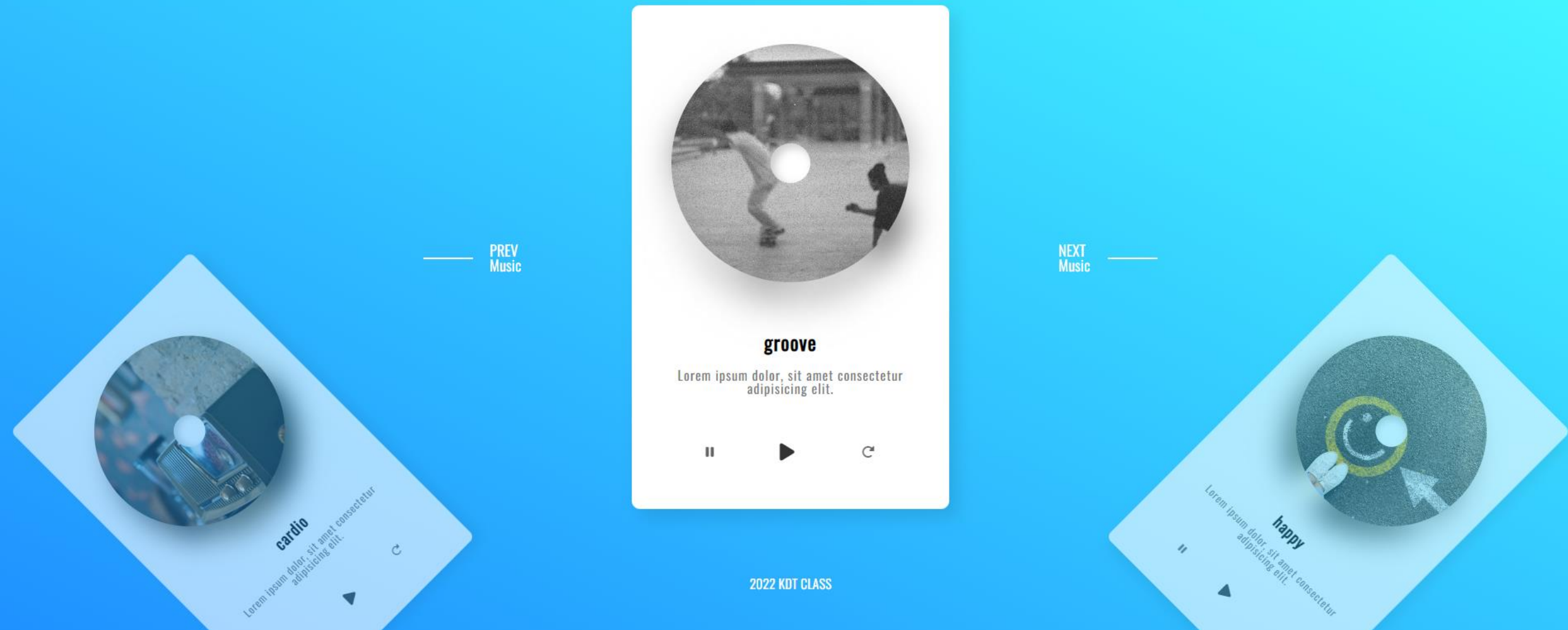
Git-ignore 가 안먹힐 때!?



- Git cache 때문에 그렇습니다!
- 보통 기존에 깃으로 관리 되던 곳에 git-ignore 를 추가하면 관리하던 파일 리스트를 git cache 가 가지고 있기 때문에 해당 문제가 발생합니다!
- `Git rm -r --cached .`
- 명령어로 깃 캐쉬를 삭제 후, 다시 `git add .` → `git commit` 하면 적용됩니다!



뮤직 플레이어 제작!



<https://tetz-music-player.netlify.app/>



좌우 이동 버튼

클릭 액션 처리

버튼, JS 처리!



- 버튼을 각각 불러옵시다!
- 해당 패널의 위치 값을 저장할 변수 만들기!
- 각각 버튼에 이벤트 붙이기!
- 0을 기준으로 next 를 누르면 $+1 * 45\text{deg}$ 씩 돌아가고
- Prev 를 누르면 $-1 * 45\text{deg}$ 씩 돌아가면 됩니다!
- 다만, 해당 속성 값은 누적이 아니므로 next 가 2번 눌리면 45deg 에서 90deg 로 돌아야 합니다!
- 즉, 해당 패널의 위치 값을 0 을 기준으로 정해서 곱하면 됩니다!



```
const prev = document.querySelector(".btnPrev");
const next = document.querySelector(".btnNext");
let num = 0;

prev.addEventListener("click", function (e) {
  frame.style.transform = `rotate(${deg * ++num}deg)`;
});

next.addEventListener("click", function (e) {
  frame.style.transform = `rotate(${deg * --num}deg)`;
});
```

- 돌아가는 애니메이션을 줘야 하는데, 돌아가는 UI는 Section 이므로 Section 에 transition: 1s; 값 추가

```
section {
  transition: 1s;
```



가운데 패널에 활성화 주기

활성화 패널, on 클래스 부여



- 첫번째 패널인 Article 에 on 클래스를 부여

```
<section>
  <article class="on">
    <div class="inner">
```

- On 클래스를 가진 패널은
 - Opacity: 1 / transform: scale(1) 부여
 - 나머지 .inner 에는 scale(0.8) 부여

```
&.on {
  .inner {
    opacity: 1;
    transform: scale(1);
  }
}
```

```
.inner {
  transform: scale(0.8);
}
```

활성화 패널, JS 처리하기



- 전역 변수로 활성화 패널의 위치를 저장 → `let active = 0;`
- Prev 버튼 처리
 - 초창기 위치가 0
 - 0에서 prev 가 눌리면? → 제일 마지막(article 의 개수 - 1)으로 이동
 - 그 외에는 자신의 값에서 -1 만 해주면 됩니다!
 - 그리고 현재 모든 article 에서 on 이라는 클래스를 제거 후 → active 위치에
만 on 클래스 추가!



```
let num = 0;
let active = 0;

prev.addEventListener("click", function (e) {
  frame.style.transform = `rotate(${deg * ++num}deg)`;

  if (active === 0) {
    active = len - 1;
  } else {
    active--;
  }
  for (let el of articleArr) {
    el.classList.remove("on");
  }
  articleArr[active].classList.add("on");
});
```

활성화 패널, JS 처리하기



- Next 버튼 처리

- 이 친구는 반대로, 마지막 위치에서 next 가 눌리면 0으로 변경!
- 나머지는 클릭 되면 +1 해주면 되겠죠?
- 역시 모든 article 에서 on 클래스 제거 후 → active 위치에 on 클래스 부여



```
next.addEventListener("click", function (e) {
  frame.style.transform = `rotate(${deg * --num}deg)`;

  if (active === len - 1) {
    active = 0;
  } else {
    active++;
  }

  for (let el of list) {
    el.classList.remove("on");
  }
  list[active].classList.add("on");
});
```




패널 꾸미기

앨범 사진, HTML 작업



- 먼저 앨범 사진을 넣어 보아요!
- 사진을 넣을 div 요소 pic 삽입!
- CD 처럼 보이기 위한 가운데 구멍 dot 도 삽입!

```
<section>
  <article class="on">
    <div class="inner">
      <div class="pic">
        <div class="dot"></div>
      </div>
    </div>
  </article>
```

앨범 사진, CSS 작업



- .pic
 - Width: 15vw / height: 15vw;
- 실제 사진은 ::before 와 ::after 를 사용해서 처리 합니다!
 - Content: ""
 - Display: block / width: 100% / height: 100%
 - position: absolute / top: 0px / left: 0px / border-radius: 50%
 - background-repeat: no-repeat / background-position: center / background-size: cover
 - 이미지 주소로 이미지 삽입!

```
section {
  article {
    .inner {
      .pic {
        height: 15vw;
        width: 15vw;
        position: relative;

        &::before,
        &::after {
          content: "";
          display: block;
          position: absolute;
          width: 100%;
          height: 100%;
          top: 0px;
          left: 0px;
          border-radius: 50%;
          background-repeat: no-repeat;
          background-position: center;
          background-size: cover;
          background-image: url("../img/cardio.jpg");
        }
      }
    }
  }
}
```



앨범 사진, CSS 작업



- 지금은 사진 2장이 겹쳐진 상태이죠?
- ::before 만 따로 이미지를 처리해서 자연스러운 그림자 효과를 줍시다
 - Transform: translste(0, 10%) / filter: blur(20px) brightness(130%);
- .dot 가운데 구멍도 처리해 보죠
 - Width: 2.5vw / height: 2.5vw / postion: absolute / top: 50% / left: 50% / transform: translate(-50%, -50%) / background-color: #fff / border-radius: 50% / box-shadow: inset 5px 5px 10px rgba(0, 0, 0, 0.2); / z-index: 3



```
section {
  article {
    .inner {
      .pic {

        &::before {
          transform: translateY(10%);
          filter: blur(20px) brightness(130%);
        }

        .dot {
          position: absolute;
          width: 2.5vw;
          height: 2.5vw;
          top: 50%;
          left: 50%;
          transform: translate(-50%, -50%);
          background-color: #fff;
          border-radius: 50%;
          box-shadow: inset 5px 5px 10px rgba(0, 0, 0, 0.2);
          z-index: 3;
        }
      }
    }
  }
}
```

앨범 사진, JS로 일괄 삽입을 해봅시다!



- 현재 앨범 사진과 음악 mp3 파일명은 동일한 상태이죠?
- JS에서 배열을 사용해서 일괄적으로 이미지를 삽입해 BoA요!
 - 각각의 파일명이 담긴 배열 만들기
 - 이미 사용한 반복문이 있으니, 해당 반복문에 처리를 해봅시다!



```
const names = ["cardio", "groove", "happy", "light", "lily", "limes", "pop", "swing"];

for (let i = 0; i < len; i++) {
  articleArr[i].style.transform = `rotate(${deg * i}deg) translateY(-100vh)`;

  // 사진 이미지 일괄 적용
  const pic = list[i].querySelector(".pic");
  pic.style.backgroundImage = `url(../img/${names[i]}.jpg)`;
}
```




앨범 사진, before / after 에 상속 시키기



- 지금은 .pic 에 사진이 들어간 상태입니다!
- .pic 사진은 필요 없으니 크기를 0 으로 만들어서 날립니다!
 - Background-size: 0;
- 그리고 ::before ::after 에 .pic 의 사진 위치를 상속 시킵니다!
 - Background-image: inherit;



```
.pic {  
  height: 15vw;  
  width: 15vw;  
  position: relative;  
  background-size: 0;  
  
  &::before,  
  &::after {  
    content: "";  
    display: block;  
    position: absolute;  
    width: 100%;  
    height: 100%;  
    top: 0px;  
    left: 0px;  
    border-radius: 50%;  
    background-repeat: no-repeat;  
    background-position: center;  
    background-size: cover;  
    background-image: inherit;  
  }  
}
```

앨범 제목, HTML 영역



- .text 영역 만들기

- H2

- P

```
<section>
  <article class="on">
    <div class="inner">
      <div class="pic">
        <div class="dot"></div>
      </div>
      <div class="text">
        <h2></h2>
        <p>Lorem ipsum dolor, sit amet consectetur.</p>
      </div>
    </div>
  </article>
```

앨범 제목, JS로 제목도 일괄 적용



- 이미 선언한 이름 배열을 또 쓰면 되겠죠?

```
const names = ["cardio", "groove", "happy", "light", "lily", "limes", "pop", "swing"];

for (let i = 0; i < len; i++) {
  articleArr[i].style.transform = `rotate(${deg * i}deg) translateY(-100vh)`;

  // 사진 이미지 일괄 적용
  const pic = articleArr[i].querySelector(".pic");
  pic.style.backgroundImage = `url(../img/${names[i]}.jpg)`;

  // 음악 제목 일괄 적용
  const title = articleArr[i].querySelector(".text>h2");
  title.innerHTML = `${names[i]}`;
}
```

앨범 제목, text 영역 CSS 작업



- **.text**
 - Width: 15vw / text-align: center
 - Position: absolute / margin-top: 60px / letter-spacing: 1px;
- **H2**
 - Margin-bottom: 20px;
- **P**
 - Color: #777;



```
section {  
  article {  
    .inner {  
  
      .text {  
        position: absolute;  
        width: 15vw;  
        text-align: center;  
        margin-top: 60px;  
        letter-spacing: 1px;  
        h2 {  
          margin-bottom: 20px;  
        }  
        p {  
          color: #777;  
        }  
      }  
    }  
  }  
}
```



음악 플레이어, HTML 작업

- UI 태그 + icon 으로 음악 플레이어를 만듭시다!
- UI > li * 3
 - 각각 li 에 icon 으로 버튼 주기

```
<div class="text">
  <h2></h2>
  <p>Lorem ipsum dolor, sit amet consectetur adipisicing elit.</p>
  <ul class="control">
    <li class="pause"><i class="fas fa-pause"></i></li>
    <li class="play"><i class="fas fa-play"></i></li>
    <li class="reload"><i class="fas fa-redo-alt"></i></li>
  </ul>
</div>
```


음악 플레이어, CSS 작업



- UI
 - 가로 배치죠? → `display: flex`
 - 버튼이 적절히 퍼지도록 → `justify-content: space-around`
- UI li
 - `Cursor: pointer;`
 - `Opacity: 0.6 / transition: 0.5s`
- `&.play`
 - 플레이 버튼은 더 크고 선명하게 보이도록 → `transform: scale(1.5) / opacity: 0.8;`

```
section {  
  article {  
    .inner {  
      .text {  
  
        .control {  
          display: flex;  
          justify-content: space-around;  
          margin-top: 60px;  
  
          li {  
            cursor: pointer;  
            opacity: 0.6;  
            transition: 0.5s;  
  
            &.play {  
              transform: scale(1.5);  
              opacity: 0.8;  
            }  
          }  
        }  
      }  
    }  
  }  
}
```



음악 플레이어, CSS 작업



- Hover 효과 주기
 - 각각의 버튼 → 1.5배 / opacity: 1
 - Play 버튼 → 2배

```
li {  
  
    &:hover {  
        transform: scale(1.5);  
        opacity: 1;  
  
        &.play {  
            transform: scale(2);  
        }  
    }  
}
```



앨범 이미지 회전

앨범 회전, Keyframes 작업



- 앨범은 ::before 와 ::after 2개의 이미지가 있으므로 2개의 keyframes 작업 필요
- 하나는 자신의 위치에서 360deg 돌고
- 하나는 자신의 위치 10% 아래에서 360deg 돌면 되므로!



```
@keyframes rotation {  
  0% {  
    transform: rotate(0deg);  
  }  
  
  100% {  
    transform: rotate(360deg);  
  }  
}  
  
@keyframes rotation2 {  
  0% {  
    transform: translateY(10%) rotate(0deg);  
  }  
  
  100% {  
    transform: translateY(10%) rotate(360deg);  
  }  
}
```

앨범 회전, CSS 작업



- 사진 요소에 on 이라는 클래스가 부여되면 회전 하도록 설정!

```
.pic {  
  &.on {  
    &::before {  
      animation: rotation2 4s linear infinite;  
    }  
  
    &::after {  
      animation: rotation 4s linear infinite;  
    }  
  }  
}
```

앨범 회전, JS 작업



- Play 버튼을 누르면 on 클래스가 부여 되도록 처리!
- Pause 버튼을 누르면 on 클래스가 제거 되도록 처리!



// CD 사진 회전

```
for (let el of articleArr) {  
  const play = el.querySelector(".play");  
  const pause = el.querySelector(".pause");  
  const load = el.querySelector(".reload");  
  
  play.addEventListener("click", function (e) {  
    e.currentTarget.closest("article").querySelector(".pic").classList.add("on");  
  });  
  
  pause.addEventListener("click", function (e) {  
    e.currentTarget.closest("article").querySelector(".pic").classList.remove("on");  
  });  
}
```



음악 파일

삽입 및 재생

음악 파일, JS로 일괄 삽입을 해봅시다!



- 아까 쓰던 반복문이 있네요!?
- createElement 로 audio 엘리먼트 만들기
- setAttribute로 위치(src) 속성 입력하기
- 재생이 무한히 반복해야 하므로 loop 속성도 주기
- 각각의 list 요소에 audio 붙여넣기!



```
for (let i = 0; i < len; i++) {  
  list[i].style.transform = `rotate(${deg * i}deg) translateY(-100vh)`;  
  
  const pic = list[i].querySelector(".pic");  
  pic.style.backgroundImage = `url(../img/${names[i]}.jpg)`;  
  
  const title = list[i].querySelector(".text>h2");  
  title.innerHTML = `${names[i]}`;  
  
  // 음악 파일 일괄 적용  
  const audio = document.createElement("audio");  
  audio.setAttribute("src", `../music/${names[i]}.mp3`);  
  audio.setAttribute("loop", "loop");  
  articleArr[i].append(audio);  
}
```

음악 파일, JS로 재생 처리!



- 아까 play 랑 pause 버튼 작업 했었죠!? 거기로 고고싱
- 재생 버튼을 클릭하면 audio 태그를 선택 후 play() 메소드로 재생
- 멈춤 버튼을 클릭하면 audio 태그를 선택 후 pause() 메소드로 멈춤
- 이번에는 reload 버튼도 작업 합시다!
 - 재생과 완전히 동일하지만 처음부터 틀어야 하므로
 - Load() 로 음악을 다시 로드 시킨 후 → play()

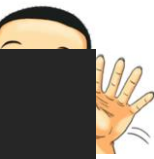
```
for (let el of articleArr) {
  const play = el.querySelector(".play");
  const pause = el.querySelector(".pause");
  const load = el.querySelector(".reload");

  play.addEventListener("click", function (e) {
    e.currentTarget.closest("article").querySelector(".pic").classList.add("on");
    e.currentTarget.closest("article").querySelector("audio").play();
  });

  pause.addEventListener("click", function (e) {
    e.currentTarget.closest("article").querySelector(".pic").classList.remove("on");
    e.currentTarget.closest("article").querySelector("audio").pause();
  });

  load.addEventListener("click", function (e) {
    e.currentTarget.closest("article").querySelector(".pic").classList.add("on");

    e.currentTarget.closest("article").querySelector("audio").load();
    e.currentTarget.closest("article").querySelector("audio").play();
  });
}
```



음악 파일, 여러 개가 한꺼번에 재생?



- 다른 노래가 재생 되면 다른 노래가 멈추도록 작업해 봅시다!!
- 이전 노래의 위치를 저장할 before 라는 전역 변수 생성!
- Before 의 초기 값을 0으로 세팅하고 before 가 0일 경우 현재 패널의 위치를 저장
- 다른 패널에서 재생 또는 Reload 버튼을 눌렀을 경우 이전 위치의 노래를 정지
 - 현재 클릭이 된 패널의 위치와 before 를 비교하여 서로 다를 경우 before 의 노래를 정지 → 현 위치를 before 로 등록!



```
play.addEventListener("click", function (e) {  
    if (before === 0) {  
        before = e.currentTarget;  
    } else if (before !== e.currentTarget) {  
        before.closest("article").querySelector("audio").pause();  
        before.closest("article").querySelector(".pic").classList.remove("on");  
        before = e.currentTarget;  
    }  
  
    e.currentTarget.closest("article").querySelector(".pic").classList.add("on");  
  
    e.currentTarget.closest("article").querySelector("audio").play();  
});
```




```
load.addEventListener("click", function (e) {  
    if (before === 0) {  
        before = e.currentTarget;  
    } else if (before !== e.currentTarget) {  
        before.closest("article").querySelector("audio").pause();  
        before.closest("article").querySelector(".pic").classList.remove("on");  
        before = e.currentTarget;  
    }  
  
    e.currentTarget.closest("article").querySelector(".pic").classList.add("on");  
    e.currentTarget.closest("article").querySelector("audio").load();  
    e.currentTarget.closest("article").querySelector("audio").play();  
});
```



