# Promise 심화

## 목표

1. promise를 만드는 방법을 이해합니다.
2. async/await의 사용법을 익힙니다.

## Promise hell (callback hell)

비동기적인 작업을 순차적으로 수행할 때 발생하는 번잡스러움

| etch('http://localhost:3000/topics')  .then(function(response){  return response.json()  })  .then(function(topics){  console.log('topics', topics);  fetch('http://localhost:3000/topics/'+topics[0].id)  .then(function(response){  return response.json();  })  .then(function(topic){  console.log('topic', topic);  })  });  f |
| --- |

위의 코드를 아래와 같이 변환할 수 있습니다!

| response = await fetch('http://localhost:3000/topics');  topics = await response.json();  console.log(topics);  response = await fetch('http://localhost:3000/topics/'+topics[0].id);  topic = await response.json();  console.log(topic); |
| --- |

## callback 만들기

| function timer(callback, time) {  setTimeout(function() {callback(time);}, time);  } |
| --- |

## 

## promise 만들기

1. new Promise(callback)을 이용해서 프라미스를 만들 수 있습니다.
2. callback은 resolve와 reject 함수를 파라미터로 받습니다.
3. resolve 함수를 호출하면 then의 콜백이 실행됩니다.

| function timer(time){  return **new Promise(function(resolve){**  **setTimeout(function(){**  **resolve(time);**  **}, time);**  **});**  } |
| --- |

## then을 이용해서 promise를 실행

| console.log('start');  timer(1000).then(function(time){  console.log('time:'+time);  return timer(time+1000);  }).then(function(time){  console.log('time:'+time);  return timer(time+1000);  }).then(function(time){  console.log('time:'+time);  console.log('end');  }); |
| --- |

## async/await를 이용해서 promise를 실행

1. promise를 리턴하는 함수 앞에 await를 붙이면 실행될 때까지 대기합니다.
2. await는 async 함수 안에서만 사용할 수 있습니다.
3. async 함수는 promise를 암시적으로 리턴합니다. → async

| async function run(){  console.log('start');  var time = await timer(1000);  console.log('time:'+time);  time = await timer(time+1000);  console.log('time:'+time);  time = await timer(time+1000);  console.log('time:'+time);  console.log('end');  return time;  } |
| --- |

## race, all

promise를 이용해서 가장 빠른 작업의 결과를 채택하거나, 동시에 작업을 실행하고 모든 작업이 끝났을 때 결과를 수집하는 등의 작업을 할 수 있습니다.

| console.time('all')  result = await Promise.all([timer(1000), timer(2000), timer(3000)]);  console.timeEnd('all'); |
| --- |

## async/await 사례

| fetch('https://jsonplaceholder.typicode.com/posts')  .then(response => response.json())  .then(data => {  console.log(data);  }) |
| --- |
| async function run(){  let response = await fetch('https://jsonplaceholder.typicode.com/posts');  let data = await response.json();  console.log(data);  }  run(); |

## 