Week5

1. Code แบบที่หนึ่งเป็น Synchrononus

Code แบบที่สองเป็น Asynchrononus เพราะ มี settimeout

|  |  |
| --- | --- |
| Callback Hell | Promise |
| const fn1 = (value, cb) => {  console.log('fn1', value);  cb(value + 1);  };  const fn2 = (value, cb) => {  console.log('fn2', value);  cb({  a2: value + 1,  b2: value + 2,  });  };  const fn3 = (value1, value2, cb) => {  console.log('fn3', value1, value2);  cb(value1 + value2);  };  const fn4 = (value, cb) => {  console.log('fn4', value);  cb();  };  const main = () => {  const a = 0;  fn1(a, (a1) => {  fn2(a1, ({ a2, b2 }) => {  fn3(a2, b2, (a3) => {  fn4(a3, () => {  console.log('End');  });  });  });  });  };  main(); | const fn1 = (value) => new Promise((resolve) => {  console.log('fn1', value);  resolve(value + 1);  });  const fn2 = (value) => new Promise((resolve) => {  console.log('fn2', value);  resolve({  a2: value + 1,  b2: value + 2  });  });  const fn3 = (value1, value2) => new Promise((resolve) => {  console.log('fn3', value1, value2);  resolve(value1 + value2);  });  const fn4 = (value) => new Promise((resolve) => {  console.log('fn4', value);  resolve();  });  const main = () => {  const a = 0;  fn1(a).then((a1) => fn2(a1))  .then(({ a2, b2 }) => fn3(a2, b2))  .then((a3) => fn4(a3))  .then(() => console.log('End'));  };  main(); |
| Promise | Await |
| const fn1 = (value) => new Promise((resolve) => {  console.log('fn1', value);  resolve(value + 1);  });  const fn2 = (value) => new Promise((resolve) => {  console.log('fn2', value);  resolve({  a2: value + 1,  b2: value + 2  });  });  const fn3 = (value1, value2) => new Promise((resolve) => {  console.log('fn3', value1, value2);  resolve(value1 + value2);  });  const fn4 = (value) => new Promise((resolve) => {  console.log('fn4', value);  resolve();  });  const main = () => {  const a = 0;  fn1(a).then((a1) => fn2(a1))  .then(({ a2, b2 }) => fn3(a2, b2))  .then((a3) => fn4(a3))  .then(() => console.log('End'));  };  main(); | function fn1(value){  console.log('fn1', value);  var value=value+1;  return value  }  function fn2(value){  console.log('fn2', value);  var value2=value+1  var value3=value+2  return [value2,value3]  }  function fn3(value){  console.log('fn3', value[0], value[1]);  value4= value[0]+value[1]  return value4  }  function fn4(value){  console.log('fn4', value);  return null  }  async function main(a){  const x = await fn1(a);  const y = await fn2(x);  const z = await fn3(y);  const zz = await fn4(z);  console.log('End')  }  console.log(main(0)) |