1. Lab. JavaScript không đồng bộ
   1. Sync

document.querySelector('button.sync').onclick = function() {

clearOutput();

output("BEGIN sync");

let ret = sFunction();

output(ret);

output("END.");

};

|  |  |  |  |
| --- | --- | --- | --- |
| Call Stack | Microtask | Macrotask | Giải thích |
| output("BEGIN sync"); |  |  | In ra “BEGIN sync” |
| sFunction() |  |  | Hàm đồng bộ → trả ngay Hello S! |
| output("Hello S!") |  |  | In ra “Hello S!” |
| output("END.") |  |  | In ra “END.” |

* 1. settimeout

document.querySelector('button.settimeout').onclick = function() {

clearOutput();

output("BEGIN settimeout");

setTimeout(

function() {

let ret = sFunction();

output(ret);

},

0);

output("END.");

};

|  |  |  |  |
| --- | --- | --- | --- |
| Call Stack | Microtask | Macrotask | Giải thích |
| output("BEGIN settimeout"); |  |  | In ra “BEGIN settimeout” |
| setTimeout(..., 0) |  | setTimeout(..., 0) | Callback được đưa vào macrotask queue |
| output("END.") |  | setTimeout(..., 0) | In ra “END.” |
| Rỗng |  | setTimeout(..., 0) | Stack trống → event loop lấy macrotask ra |
| output("Hello S!") |  |  | In ra “Hello S!” |

* 1. promise

document.querySelector('button.promise').onclick = function() {

clearOutput();

output("BEGIN promise");

aFunction1().then(function (ret) {

output(ret);

});

output("END.");

};

|  |  |  |  |
| --- | --- | --- | --- |
| Call Stack | Microtask | Macrotask | Giải thích |
| output("BEGIN promise"); |  |  | In “BEGIN promise” |
| aFunction1() | Then(...) |  | Promise resolve ngay → callback .then() đưa vào microtask queue |
| output("END."); | Then(...) |  | In “END.” |
| rỗng | Then(...) |  | Call stack trống → chạy microtask |
| output(“Hello A1!”) |  |  | In kết quả |

* 1. async

document.querySelector('button.async').onclick = function() {

clearOutput();

output("BEGIN async");

aFunction2().then(function (ret) {

output(ret);

});

output("END.");

};

|  |  |  |  |
| --- | --- | --- | --- |
| Call Stack | Microtask | Macrotask | Giải thích |
| output("BEGIN async"); |  |  | In “BEGIN async” |
| aFunction2() | Then(...) |  | Promise resolved → microtask |
| output("END."); | Then(...) |  | In “END.” |
| Rỗng | Then(...) |  | Chạy microtask |
| output("Hello A2!") |  |  | In kết quả |

* 1. await

document.querySelector('button.await').onclick = async function() {

clearOutput();

output("BEGIN await");

let ret = await aFunction1();

output(ret);

output("END.");

};

|  |  |  |  |
| --- | --- | --- | --- |
| Call Stack | Microtask | Macrotask | Giải thích |
| onclick (async) |  |  |  |
| output("BEGIN await"); |  |  | In “BEGIN await” |
| await aFunction1() | aFunction1()  output("Hello A1")  output("END.") |  | Promise resolve ngay, phần sau await được đưa vào microtask queue |
| rỗng | aFunction1()  output("Hello A1")  output("END.") |  | callback trống thực hiện trong microtask |
| aFunction1()  output("Hello A1")  output("END.") |  |  | In kết quả |

* 1. sync vs async

document.querySelector('button.syncvsasync').onclick = function() {

clearOutput();

output("BEGIN sync vs async");

aFunction1().then(function (ret) {

output(ret);

});

aFunction2().then(function (ret) {

output(ret);

});

let ret = sFunction(1);

output(ret);

output("END.");

};

|  |  |  |  |
| --- | --- | --- | --- |
| Call Stack | Microtask | Macrotask | Giải thích |
| output("BEGIN sync vs async"); |  |  | In “BEGIN...” |
| aFunction1() | Then(output("Hello A1!")) |  | Promise resolved → microtask |
| aFunction2() | Then(output("Hello A1!"))  then(output("Hello A2!")) |  | Promise resolved → microtask |
| sFunction() | ... |  | Đồng bộ, trả "Hello S!" |
| output("Hello S!") | ... |  | In “Hello S!” |
| output("END.") | ... |  | In “END.” |
| trống | Then(output("Hello A1!"))  then(output("Hello A2!")) |  | Thực hiện microtasks theo thứ tự |
| output("Hello A1!") | then(output("Hello A2!")) |  | In A1 |
| output("Hello A2!") |  |  | In A2 |

* 1. waitall

document.querySelector('button.waitall').onclick = function() {

clearOutput();

output("BEGIN waitall");

let p1 = aFunction1();

let p2 = aFunction2();

Promise.all([p1, p2]).then(function (arr) {

output(arr[0]);

output(arr[1]);

});

output("END.");

};

|  |  |  |  |
| --- | --- | --- | --- |
| Call Stack | Microtask | Macrotask | Giải thích |
| output("BEGIN waitall"); |  |  | In “BEGIN...” |
| aFunction1()  aFunction2() | Promise.all([p1, p2]).then(...) |  | Tất cả Promise resolve, nhưng .then() nằm trong microtask |
| output("END.") | .then(...) |  | In “END.” |
| rỗng | .then(...) |  | Chạy microtask Promise.all |
| output("Hello A1!"), output("Hello A2!") |  |  | In kết quả |

* 1. mixed

document.querySelector('button.mixed').onclick = function() {

clearOutput();

output("BEGIN mixed");

setTimeout(() => { output('setTimeout 1'); }, 10);

setTimeout(() => { output('setTimeout 2'); }, 0);

new Promise(function(fulfill, reject) { fulfill('Promise 1'); })

.then (function(res) { output(res); return "Sub Promise 1";})

.then (function(res) { output(res); });

new Promise((fulfill, reject) => { fulfill('Promise 2'); })

.then (res => {output(res); return "Sub Promise 2";})

.then (res => {output(res);});

output("END.");

};

|  |  |  |  |
| --- | --- | --- | --- |
| Call Stack | Microtask | Macrotask | Giải thích |
| output("BEGIN mixed"); |  |  | In “BEGIN...” |
| setTimeout(..., 10); |  | setTimeout(..., 10); | Thêm vào macrotask |
| setTimeout(..., 0); |  | setTimeout0  setTimeout10 | Thêm vào macrotask |
| new Promise 1 | Then 1.1, 1.2 | setTimeout0  setTimeout10 | Promise resolve → thêm microtask |
| new Promise 2 | Then 1.1, 1.2  then 2.1, 2.2 | setTimeout0  setTimeout10 | Promise resolve → thêm microtask |
| output("END."); | ... | ... | In “END.” |
| Trống | ... | ... | Chạy toàn bộ microtask trước |
| output("Promise 1") | Then 1.2, 2.1, 2.2 | ... | In Promise 1 |
| output("Sub Promise 1") | Then 2.1, 2.2 | ... | In Sub Promise 1 |
| output("Promise 2") | 2.2 | ... | In Promise 2 |
| output("Sub Promise 2") |  | ... | In Sub Promise 2 |
| Trống |  | setTimeout0  setTimeout10 | Bắt đầu chạy macrotask |
| output("setTimeout 2") |  | setTimeout10 | In “setTimeout 2” |
| output("setTimeout 1") |  |  | In “setTimeout 1” |

1. Lab. Web worker
   1. Dedicated worker

|  |  |
| --- | --- |
| **Hành động** | **Kết quả** |
| Bấm “Increase” ở iframe 1 | Chỉ bộ đếm trong iframe 1 tăng |
| Bấm “Increase” ở iframe 2 | Chỉ bộ đếm trong iframe 2 tăng |
| Tổng kết | Mỗi iframe có **bộ đếm riêng biệt**, bắt đầu từ 0 |

* 1. Shared worker

|  |  |
| --- | --- |
| **Hành động** | **Kết quả** |
| Bấm “Increase” ở iframe 1 | Bộ đếm tăng và phản hồi **cập nhật ở cả hai iframe** |
| Bấm “Increase” ở iframe 2 | Bộ đếm tăng tiếp từ giá trị trước đó |
| Tổng kết | Cả hai iframe **dùng chung một biến đếm** trong cùng worker |