
node
JS

NESTED CALLBACK

ARROW FUNCTIONS

```
function sum(a, b) {
    return a + b;
}
```

=

```
const sum = (a, b) => a + b;
```

```
1  const someFunction = (arg1, callback) => {
2      // ... some code
3      console.log('someFunction', arg1);
4      callback('A');
5  };
6
7  const secondFunction = (arg1, arg2, secondCallback) => {
8      console.log('secondFunction', arg1, arg2);
9      secondCallback();
10 };
11
12 const anotherFunction = (anotherCallback) => {
13     console.log('anotherFunction');
14     anotherCallback();
15 };
16
17
18 someFunction('abc', (resultSomeFunction) => {
19     secondFunction(resultSomeFunction, 'B', () => {
20         anotherFunction(() => {
21             console.log('anotherCallback call');
22         });
23     });
24 });
```

File Edit Selection View Go Debug Terminal Help app.js - node - Visual Studio Code

app.js

app.js > ...

```js

const someFunction = (arg1, callback) => {
    // ... some code
    console.log('someFunction', arg1);
    callback('A');
};

const secondFunction = (arg1, arg2, secondCallback) => {
    console.log('secondFunction', arg1, arg2);
    secondCallback();
};

const anotherFunction = (anotherCallback) => {
    console.log('anotherFunction');
    anotherCallback();
};


someFunction('abc', (resultSomeFunction) => {
    secondFunction(resultSomeFunction, 'B', () => {
        anotherFunction(() => {
            console.log('anotherCallback call');
        });
    });
});
```

TIME TO DEBUG!

⊗ 0 ⚠ 0 Ln 25, Col 4 Spaces: 4 UTF-8 CRLF JavaScript

File Edit Selection View Go Debug Terminal Help hell.js - live - Visual Studio Code

hell.js

```javascript
// CALLBACK HELL!

getUser(username, password, (user) => {
    getLocation(user.id, (location) => {
        getWeather(location.geo, (weather) => {
            getWeatherImage(weather.type, (image) => {
                saveWeatherImage(image, (error) => {
                    if (!error) {
                        console.log('Image was saved');
                        // ...
                    }
                });
            });
        });
    });
});
```

master 0 0 Ln 1, Col 18 Spaces: 4 UTF-8 CRLF JavaScript

CALLBACK HELL!


THE PYRAMID OF DOOM

File Edit Selection View Go Debug Terminal Help hell.js - live - Visual Studio Code

hell.js

```javascript
// CALLBACK HELL!

getUser(username, password, (user) => {
    getLocation(user.id, (location) => {
        getWeather(location.geo, (weather) => {
            getWeatherImage(weather.type, (image) => {
                saveWeatherImage(image, (error) => {
                    if (!error) {
                        console.log('Image was saved');
                        // ...
                    }
                });
            });
        });
    });
});
```

master 0 0 Ln 1, Col 18 Spaces: 4 UTF-8 CRLF JavaScript

File Edit Selection View Go Debug Terminal Help lab03.js - live - Visual Studio Code

lab03.js

```javascript
const request = require('request');
const argv = require('yargs').argv;

const url = `https://jsonplaceholder.typicode.com/users/${argv.id}`;
request(url, (error, response, body) => {
    if (error) {
        console.log('błąd połączenia z adresem');
    } else if (response.statusCode !== 200) {
        console.log('nie znaleziono użytwkonika');
    } else {
        const user = JSON.parse(body);

        const lat = user.address.geo.lat;
        const lng = user.address.geo.lng;
        console.log('name:', user.name);
        console.log('lat:', lat);
        console.log('lng:', lng);
        const weatherUrl = `https://api.openweathermap.org/data/2.5/weather?appid=0ed761
        request(weatherUrl, (error, response, body) => {
            if (error) {
                console.log('błąd połączenia z adresem');
            } else if (response.statusCode !== 200) {
                console.log('nie znaleziono użytwkonika');
            } else {
                const weather = JSON.parse(body);
                console.log('temperature', weather.main.temp);
            }
        });
    }
});
```

master 0 0 Ln 30, Col 4 Spaces: 4 UTF-8 CRLF JavaScript

File Edit Selection View Go Debug promise.js - live - Visual St...

promise.js

```javascript
getUser(username, password)
    .then(user => getLocation(user.id))
    .then(location => getWeather(location.geo))
    .then(weather => getWeatherImage(weather.type))
    .then(image => saveWeatherImage(image))
    .catch(error => console.log(error));

```

master 0 0 Ln 7, Col 1 Spaces: 4 UTF-8 CRLF JavaScript 1

File Edit Selection View Go Debug Terminal Help hell.js - live - Visual Studio Code

hell.js

```javascript
// CALLBACK HELL!

getUser(username, password, (user) => {
    getLocation(user.id, (location) => {
        getWeather(location.geo, (weather) => {
            getWeatherImage(weather.type, (image) => {
                saveWeatherImage(image, (error) => {
                    if (!error) {
                        console.log('Image was saved');
                        // ...
                    }
                });
            });
        });
    });
});
```

master 0 0 Ln 1, Col 18 Spaces: 4 UTF-8 CRLF JavaScript

PROMISE

PROMISE

Klasa pozwalająca na tworzenie OBIEKTÓW reprezentujących wartość lub niepowodzenie operacji asynchronicznych.

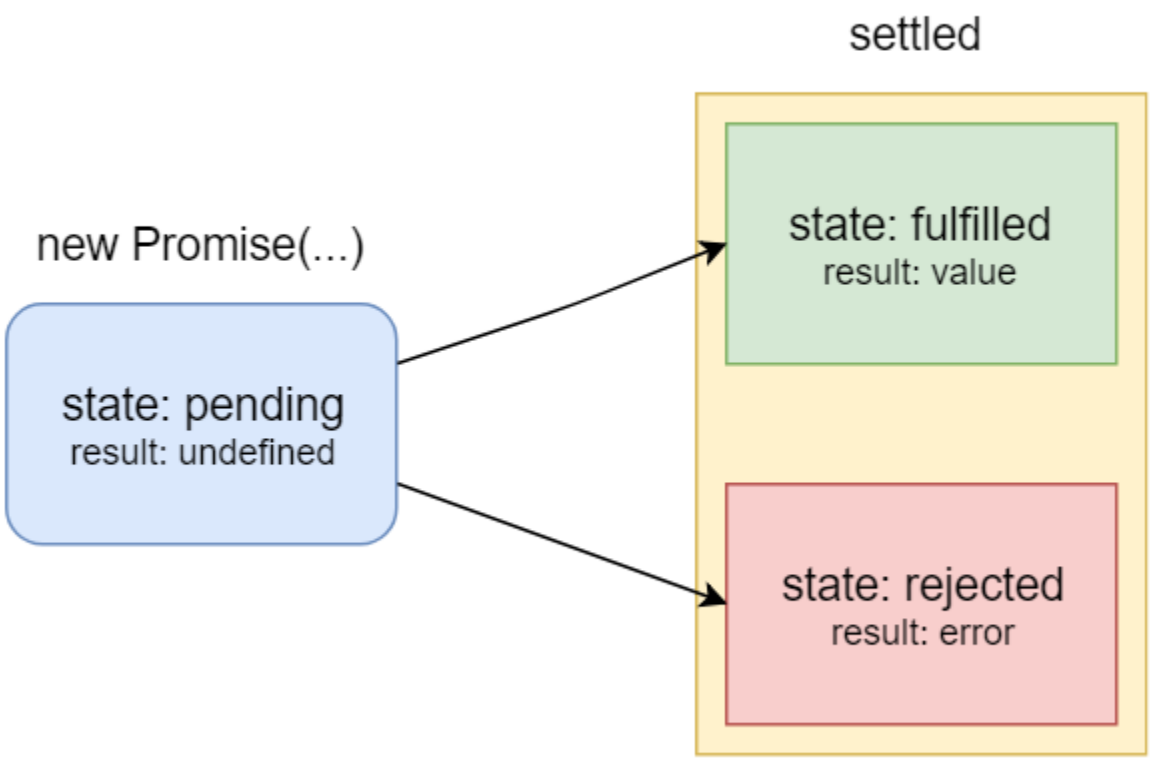
Promise reprezentuje operację która jeszcze się nie zakończyła, ale oczekuje się jej w przyszłości.

CALLBACKS VS PROMISES

- **Callback** jest funkcją, **promise** jest obiektem
- **Callback** przyjmuje parametry, **promise** jedynie zwraca wartość
- **Callback** obsługuje sukces oraz błąd, **promise** nie obsługuje nic a jedynie przekazuje dalej wartości
- **Callback** możemy wywołać wiele razy, **promise** jest wywoływany tylko raz

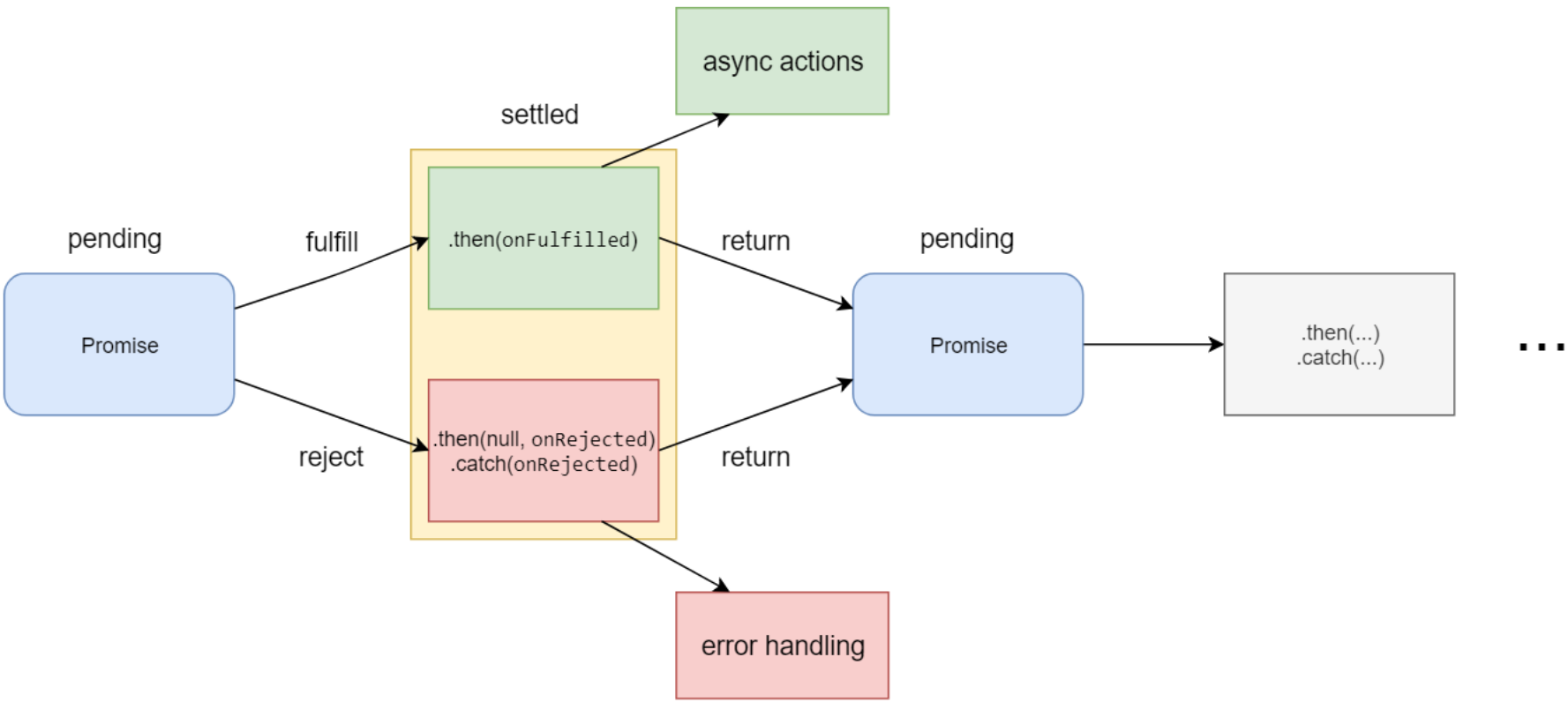
PROMISE STATES

- pending
- fulfilled
- rejected


settled
new Promise(...)
state: pending
result: undefined
state: fulfilled
result: value
state: rejected
result: error

CREATE PROMISE

```
1 const myPromise = new Promise(/* executor */ (resolve, reject) => {
2     if (/* some logic */) {
3         resolve('all works fine');
4     } else {
5         reject('error');
6     }
7 });
```


pending
Promise
fulfill
reject
settled
.then(onFulfilled)
.then(null, onRejected)
.catch(onRejected)
async actions
error handling
return
return
pending
Promise
.then(...)
.catch(...)
...

File Edit Selection View Go Debug Terminal Help • app.js - live - Visual Studio Code

DEBUG

VARIABLES

Local

this: Object

__dirname: "d:\pod...

__filename: "d:\po...

exports: Object {}

WATCH

CALL STACK PAUSED ON ...

(anonymous function)

Module._compile m...

Module._extensions..j

app.js

app.js ▸ ...

```
1
2  const myPromise = new Promise((resolve, reject) => {
3      // ...
4  });
5
6
7
8
9  console.log(myPromise);
10
```

Promise { pending }

[[PromiseStatus]]: "pending"

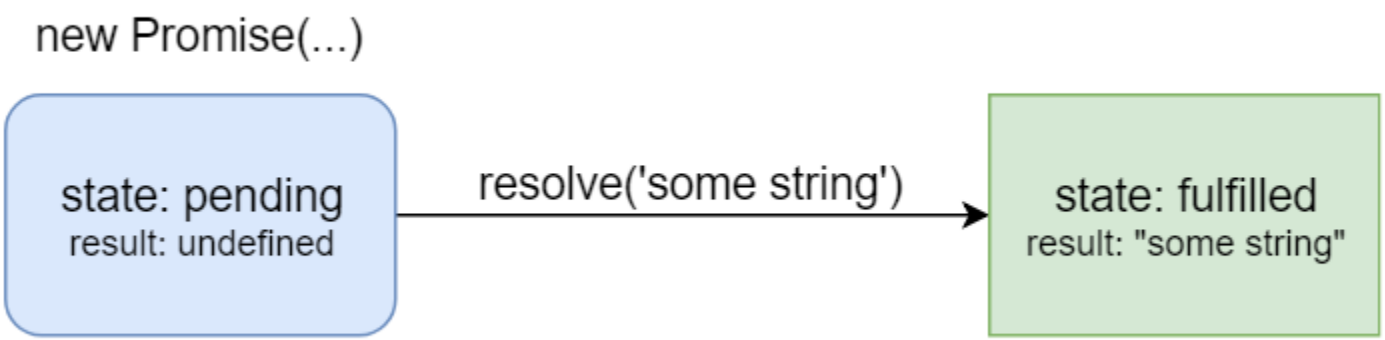
[[PromiseValue]]: undefined

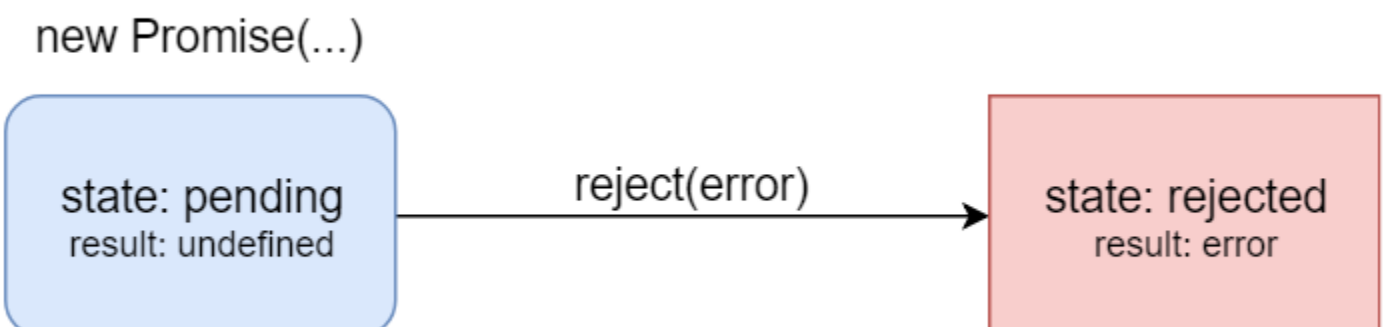
__proto__: Promise {constructor: , then: , c

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
C:\Program Files\nodejs\node.exe --inspect-brk=27222 app.js
Debugger listening on ws://127.0.0.1:27222/b1501f10-3be2-4530-bb3e-dd
0f82b811c4
```

master 0 0 Ln 8, Col 1 Spaces: 4 UTF-8 CRLF JavaScript 1


new Promise(...)
state: pending
result: undefined
resolve('some string')
state: fulfilled
result: "some string"


new Promise(...)
state: pending
result: undefined
reject(error)
state: rejected
result: error

.then(onFulfilled, onRejected)

```
1 myPromise.then(
2     result => { /* handle a succcessful result */ },
3     error => { /* handle an error */ }
4 );
```

```
1 const myPromise = new Promise((resolve, reject) => {
2     if (/* some logic */) {
3         resolve('all works fine');
4     } else {
5         reject('error');
6     }
7 });
```

```
1 myPromise.then(
2     result => {
3         console.log(result);
4     },
5     error => {
6         console.log(error);
7     }
8 );
```

.catch(onRejected)

```
1 myPromise
2     .then(result => { /* ... */ })
3     .then(result => { /* ... */ })
4     .catch(error => { /* handle an error */ });
```

```
1 const myPromise = new Promise((resolve, reject) => {
2     if (/* some logic */) {
3         resolve('all works fine');
4     } else {
5         reject('error');
6     }
7 });
```

```
1 myPromise
2     .then(result => {
3         console.log(result);
4     })
5     .catch(error => {
6         console.log(error);
7     });
```

.finally(onFinally)

```
1 myPromise
2     .then(result => { /* ... */ })
3     .then(result => { /* ... */ })
4     .catch(error => { /* handle an error */ })
5     .finally(() => { /* do something at the end ... */ });
```

Promise.all([...])

```
1  const promise1 = new Promise((resolve, reject) => {
2      // ...
3      resolve('abc');
4  });
5
6
7  const promise2 = new Promise((resolve, reject) => {
8      // ...
9      resolve('xyz');
10 });
11
12
13 Promise.all([promise1, promise2]);
```

Promise.race([...])

```
1  const promise1 = new Promise((resolve, reject) => {
2      // ...
3      setTimeout(() => resolve('abc'), 1000);
4  });
5
6
7  const promise2 = new Promise((resolve, reject) => {
8      // ...
9      setTimeout(() => resolve('xyz'), 500);
10 });
11
12
13 Promise.race([promise1, promise2]);
```

Promise.resolve(...)

```
1  Promise.resolve(/* some object/string/etc... */);
2
3
4  Promise
5      .resolve('resolved promise')
6      .then(
7          answer => console.log(answer)
8      )
9      .catch(
10         error => console.log('enter here ?', error)
11     );
```

Promise.reject(...)

```
1  Promise.reject(/* some object/string/etc... */);
2
3
4  Promise
5      .reject('some error!!!')
6      .then(
7          answer => console.log('enter here ?', answer)
8      )
9      .catch(
10         error => console.log(error)
11     );
```

ERROR HANDLING

File Edit Selection View Go Debug app.js - nodejs - Visual Studi...

JS app.js

JS app.js

```
1   JSON.parse('crash JSON!');
```

PROBLEMS TERMINAL ... 1: Microsoft.Notes.Desk

```
PS D:\nodejs> node .\app.js
```

0 0 Ln 1, Col 27 Spaces: 4 UTF-8 CRLF JavaScript

File Edit Selection View Go Debug app.js - nodejs - Visual Studi...

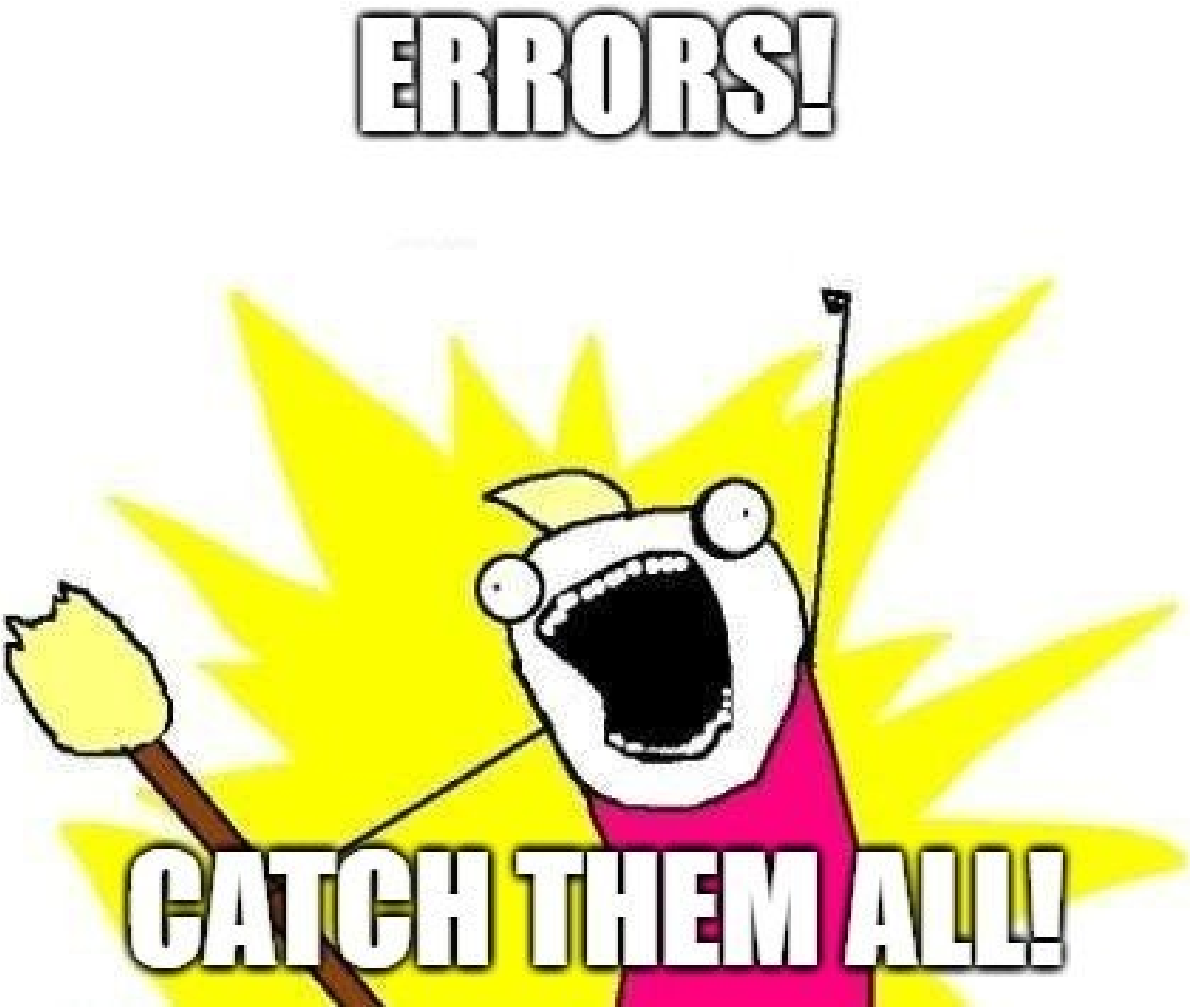
app.js

```
JSON.parse('crash JSON!');
```

PROBLEMS TERMINAL ... 1: Microsoft.Notes.Desk

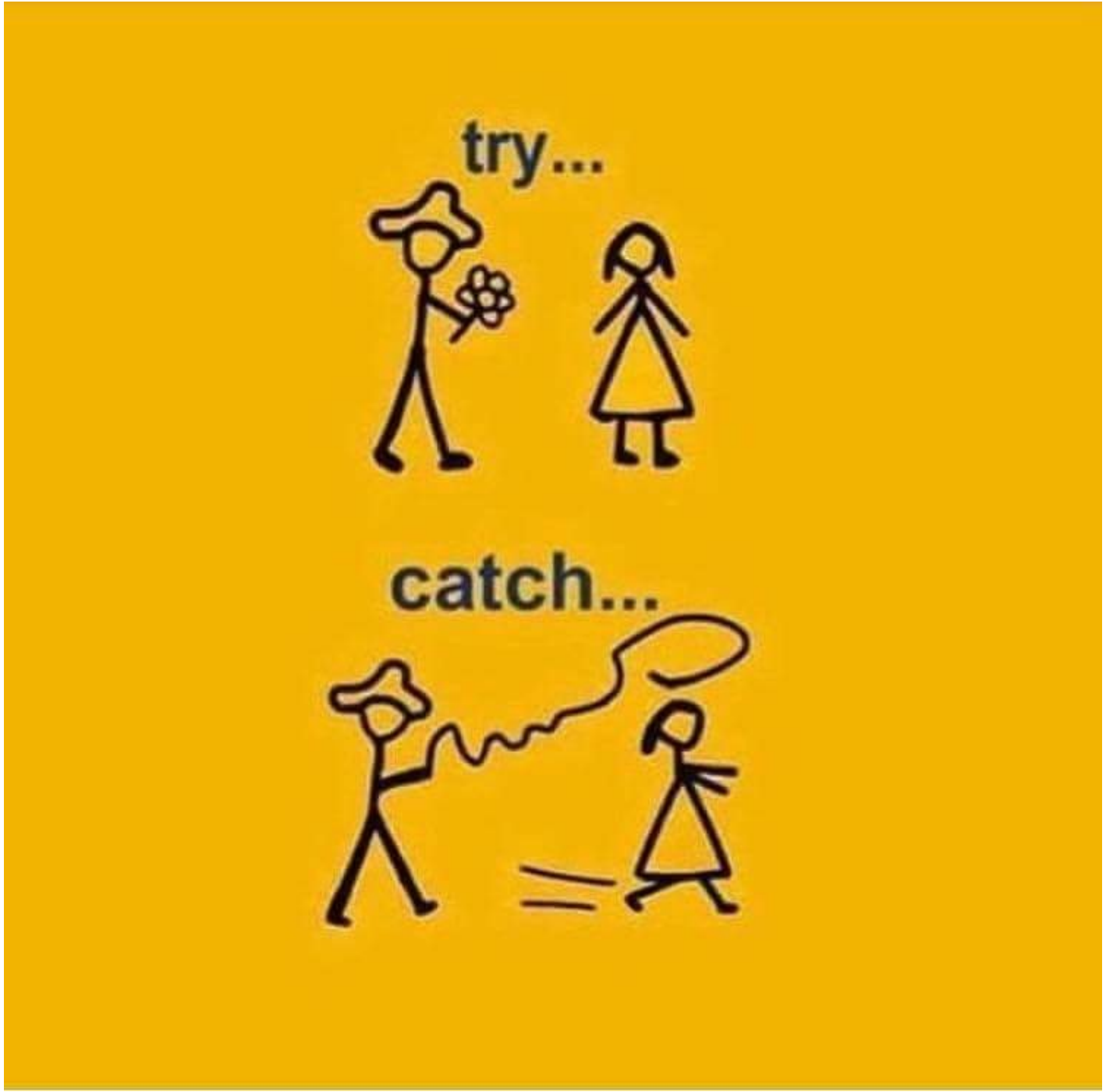
```
PS D:\nodejs> node .\app.js
undefined:1
crash JSON!
^

SyntaxError: Unexpected token c in JSON at position 0
    at JSON.parse (<anonymous>)
    at Object.<anonymous> (D:\nodejs\app.js:1:68)
    at Module._compile (internal/modules/cjs/loader.js:722:30)
    at Object.Module._extensions..js (internal/modules/cjs/loader.js:733:10)
    at Module.load (internal/modules/cjs/loader.js:620:32)
    at tryModuleLoad (internal/modules/cjs/loader.js:560:12)
    at Function.Module._load (internal/modules/cjs/loader.js:552:3)
    at Function.Module.runMain (internal/modules/cjs/loader.js:775:12)
    at startup (internal/bootstrap/node.js:300:19)
```

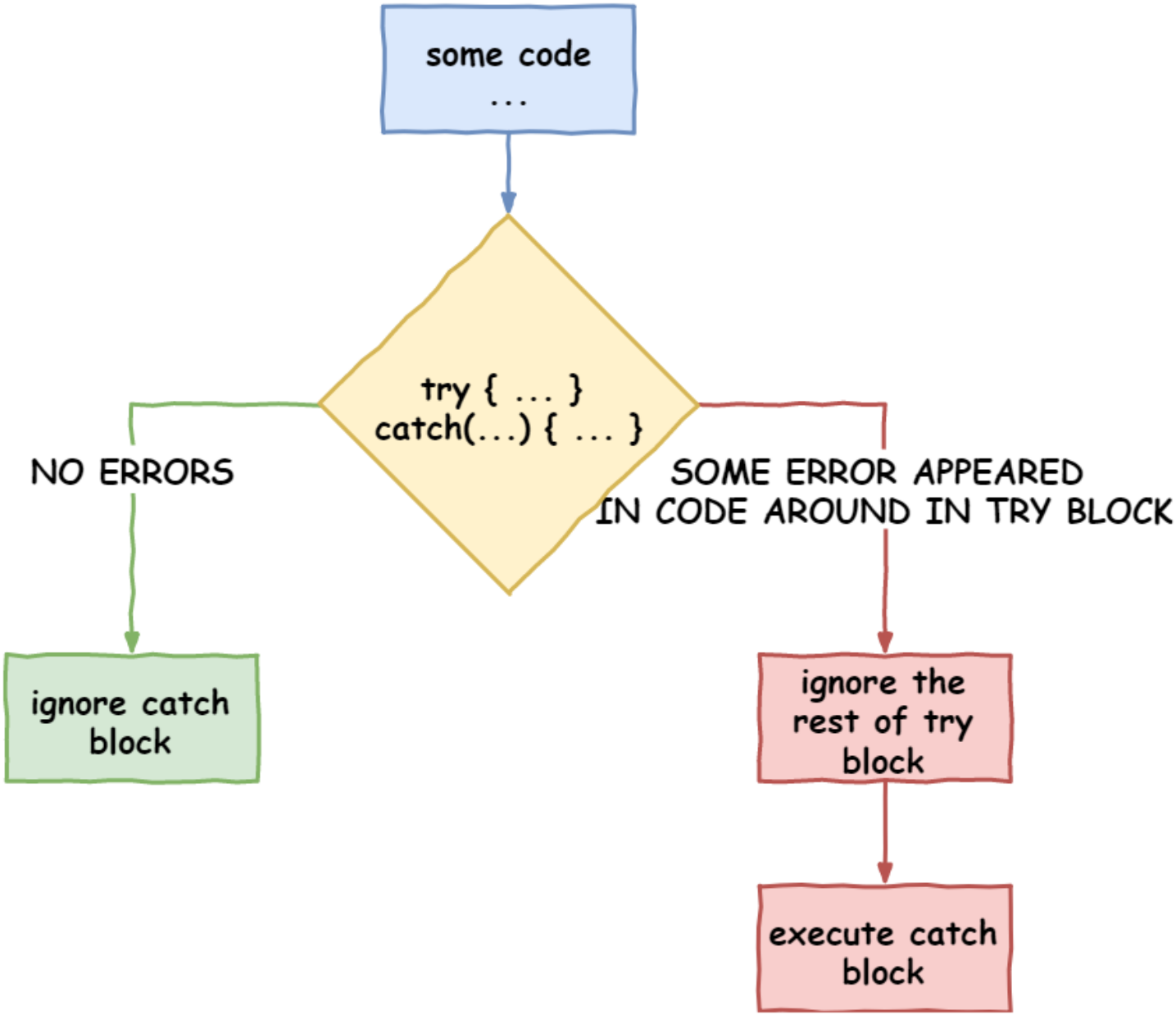
0 0 Ln 1, Col 27 Spaces: 4 UTF-8 CRLF JavaScript


ERRORS!
CATCH THEM ALL!

try ... catch

```
1 try {
2         // code...
3 } catch (error) {
4         // error handling
5 }
```


try...
catch...


some code
...
try { ... }
catch(...) { ... }
NO ERRORS
SOME ERROR APPEARED
IN CODE AROUND IN TRY BLOCK
ignore catch
block
ignore the
rest of try
block
execute catch
block

File Edit Selection View Go Debug app.js - nodejs - Visual Stu...

app.js

```js
try {
    JSON.parse('crash JSON!');
} catch (error) {
    console.log(error.message);
}
```

PROBLEMS TERMINAL ... 1: powershell

```
PS D:\nodejs> node .\app.js
Unexpected token c in JSON at position 0
PS D:\nodejs>
```

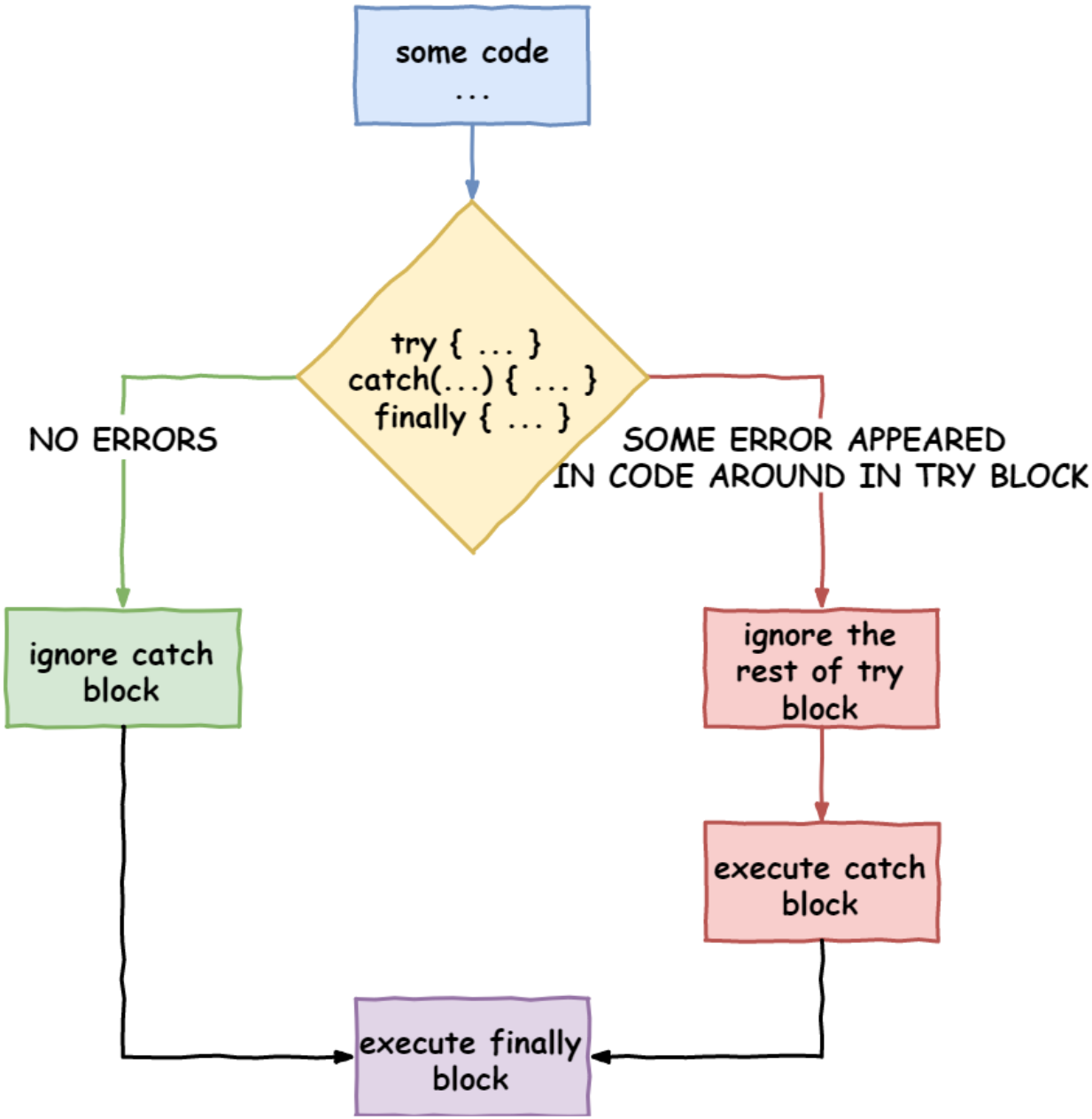
0 0 Ln 5, Col 2 Spaces: 4 UTF-8 CRLF JavaScript

try ... catch

- **try..catch** działa tylko w przypadku błędów w czasie wykonywania
- **try..catch** działa synchronicznie
- **try..catch** zmienne zadeklarowane są lokalne
- **return & finally** - **finally** wykona się zawsze nawet jak w bloku wystąpi **return**

try ... catch ... finally

```
1 try {
2         // code...
3 } catch (error) {
4         // error handling
5 } finally {
6         // some code ...
7 }
```


some code
...
try { ... }
catch(...) { ... }
finally { ... }
NO ERRORS
SOME ERROR APPEARED
IN CODE AROUND IN TRY BLOCK
ignore catch
block
ignore the
rest of try
block
execute catch
block
execute finally
block

Error types

- **Error** - bazowa klasa błędu
- **EvalError** - błąd w funkcji **eval()**
- **RangeError** - wartość z poza zakresu
- **ReferenceError** - bład referencji do obiektu
- **SyntaxError** - błąd składniowy
- **TypeError** - bład typu
- **URIError** - błąd w funkcji **encodeURI()**

Error Object

- **name** - nazwa błędu
- **message** - wiadomość błędu
- **stack** - stack w którym pojawił się błąd

EvalError

EvalError => SyntaxError

RangeError

```
1 const someNumber = 5;
2 someNumber.toPrecision(500); // this throw RangeError
```

ReferenceError

```
1 const a = 5;
2 const result = a + b; // this throw ReferenceError
```

SyntaxError

```
1 JSON.parse('crush JSON!'); // this throw SyntaxError
2 eval('const abc = "some broken string'); // this throw Syntax
```

TypeError

```
1 const someNumber = 5;
2 someNumber.toUpperCase(); // this throw TypeError
```

URIError

```
1 decodeURI('%%%'); // this throw URIError
```

Throw exception

```
// throw <error>;

throw 123;

throw "some error string";

throw new Error('some error');
```

File Edit Selection View Go Debug app.js - nodejs - Visual Studi...

app.js

app.js ▸ ...

```js
try {
    const someNumber = 3;
    if (someNumber % 2) {
        throw new Error('number is odd');
    }
    console.log(someNumber);
} catch (error) {
    console.log(error.message);
}
```

PROBLEMS TERMINAL ... 1: Microsoft.Notes.Desk

```
PS D:\nodejs> node .\app.js
number is odd
PS D:\nodejs>
```

0 0 Ln 6, Col 29 Spaces: 4 UTF-8 CRLF JavaScript

instanceof

```
1 const someError = new SyntaxError('abc');
2
3 console.log(someError instanceof ReferenceError); // false
4
5 console.log(someError instanceof SyntaxError); // true
6
7 console.log(someError instanceof Error); // true
```

PROMISE

async/await

async

```
1 async function someFunc() {
2     return ...; // some function body
3 }
```

```
1  const arrowFunc = async () => ... // some function body
```

```javascript
function multi(a, b) {
    return new Promise((resolve, reject) => {
        resolve(a * b);
    });
}
```

```javascript
async function multi(a, b) {
    return a * b;
}
```

```
1 function multi(a, b) {
2     return new Promise((resolve, reject) => {
3         reject(new Error('some error'));
4         resolve(a * b);
5     });
6 }
```

```
1 async function multi(a, b) {
2     throw new Error('some error');
3     return a * b;
4 }
```

```javascript
1 async function multi(a, b) {
2     return a + b;
3 }
4
5 multi(4, 5)
6     .then(result => {
7         console.log(result);
8     });
```

File Edit Selection View app-promise.js - node...

app-promise.js

```javascript
async function multi(a, b) {
    return a + b;
}

multi(4, 5)
    .then(result => {
        console.log(result);
    });
```

TERMINAL 1: Microsoft.Notes.Des

```
PS D:\nodejs> node .\app-promise.js
9
PS D:\nodejs>
```

0 0 Ln 6, Col 22 Spaces: 4 UTF-8 CRLF JavaScript

await

```
1 const someVar = await <promise>;
2
3
4 const someVar = await Promise.resolve('abc');
5
6
7 const axios = require('axios');
8 const user = await axios('https://.../users/1');
```

```
1 const axios = require('axios');
2
3 axios('https://jsonplaceholder.typicode.com/users/2')
4     .then((response) => {
5         console.log(response.data.name);
6     });
```

```
1 const axios = require('axios');
2
3 (async function() {
4     const response = await axios('https://jsonplaceholder.typicode.com/users/
5     console.log(response.data.name);
6 })();
```

async/await

- **async** tworzy nam funkcję opakowaną w **Promise**
- **await** działa jedynie wewnątrz funkcji **async**
- **await** nie można stosować w funkcji synchronicznej
- **await** nie działa w kodzie najwyższego poziomu

Error handling in Promise

```
1 async function someFunc() {
2     await Promise.reject(new Error('error'));
3 }
```

```
1 async function someFunc() {
2     throw new Error('error');
3 }
```

```
1 const axios = require('axios');
2
3 axios('https://jsonplaceholder.typicode.com/users/2')
4     .then((response) => {
5         console.log(response.data.name);
6     })
7     .catch(error => {
8         console.log(error)
9     });
```

```
1 const axios = require('axios');
2
3 (async function () {
4     try {
5         const response = await axios('https://jsonplaceholder.typicode.com/users/2')
6         console.log(response.data.name);
7     } catch (error) {
8         console.log(error);
9     }
10 })();
```