# **Exceptions**

## try, catch, finally statements

Javascript kalboje yra naudojami specialios paskirties kodo blokai, tam kad galėtum valgyti galimas klaidas (errors).

Klaidos Javascript kalboje galime skirstyti į dvi rūšis, sintaksės klaidas “Syntax Error” ir programos leidimo klaidos “Runtime Error”. Sintaksės klaida įvyksta kada mūsų kodas turi klaidas kurių Javascript interpretuotojas neatpažįsta. Pvz console.log('your result'; Šitas kodas išmes sintaksinę klaidą ir ji yra padarome programuotojo.

Visos kitos klaidų rūšys yra Runtime Erros, kuriuos atsiranda programai leidžiantis dėl prastai parašytos programos, neapgalvoto galimo scenarijaus arba dėl aplinkybių kurių mes kontroliuoti negalime tarkim vartotojo input, serverio klaidos.

Klaidos kurios atsiranda Runtime metu yra vadinamos “exceptions”. Ir jas galime valdyti su Javascript implementuotais būdais.

Try catch statement yra skirtas tvarkyti valdyti klaidas.

| try {  *// body of try* }  catch(error) {  *// body of catch*  } |
| --- |

Pagrindinis kodas yra try bloke, ir jei nors viena klaida jame įvyksta, programa nesustoja, tačiau ji pradeda vykdyti tai kas yra catch bloke. Jei vykdant koda try bloke klaidos nėra, catch blokas yra praleidžiamas.

| const skaicius = 10; const daliklis = 2;  try {  console.log(skaicius/daliklis);   *// pamiršome deklaruoti a kintamaji*   console.log(a); } catch(error) {  console.log('An error caught');   console.log('Error message: ' + error);  } |
| --- |

Yra dar trečias kodo blokas finally, kuris bus paleidžiamas nesvarbu ar klaida įvyko ar ne.

| try {  *// try\_statements* }  catch(error) {  *// catch\_statements*  } finally() {  *// codes that gets executed anyway* } |
| --- |

| const skaicius = 10; const daliklis = 2;  try {  console.log(skaicius/daliklis);   *// pamirsome deklaruoti a kintamaji*   console.log(a); } catch(error) {  console.log('An error caught');   console.log('Error message: ' + error);  } finally {  console.log('Finally will execute every time'); } |
| --- |

try catch blokai nepagauna klaidų jei jos yra asinchroniškos, pvz su setTimeout() funkcija.

| try {  setTimeout(function() {  *// error in the code*  }, 3000); } catch (e) {  console.log( "won't work" ); } |
| --- |

Jei norime pagauti klaidas setTimeout funkcijoje, try catch statementa turime aprašyti callback funkcijos viduje.

| setTimeout(function() {  try {  *// error in the code*  } catch {  console.log( "error is caught" );  } }, 3000); |
| --- |