

# Sysbas1A

Debuggen

# Vandaag

- Bugs en errors
- Hoe vind je bugs
- Vermijden
- Oefenen

## Be like a Programmer



John is a programmer  
John has 3 bugs to resolve,  
2 functionalities to implement,  
and needs to attend a meeting



John decides to sleep

# Bugs en errors

- Typos
- Verkeerde data type of undefined
- Out of index
- Scope
- Volgorde
- Missende functies

Coworker: so how did you solve the bug?

Me:

## Ostrich algorithm

---

From Wikipedia, the free encyclopedia

In [computer science](#), the **ostrich algorithm** is a strategy of ignoring potential problems on the basis that they may be exceedingly rare. It is named for the [ostrich effect](#) which is defined as "to stick one's head in the sand and pretend there is no problem". It is used when it is more cost-effective to allow the problem to occur than to attempt its prevention.

# Bugs en errors -> Typos

- Taalfouten
  - `elipse()` of `ellipse()` Dubbele l
- Missende hoofdletters
  - `xSpeed` of `xspeed` Hoofdletter

# Bugs en errors -> Verkeerde data type

- `"8" + 1 = ?` Een int (getal) optellen bij een string (woord) kan, maar geeft een ander antwoord dan je zou verwachten (namelijk 81)
- `"81" - 1 = ?`
- `undefined`

## Bugs en errors -> Out of index

```
let array = [0, 1, 3, 4, 1, 6, 7, 0, 2, 5]  
array[10] = ? Array index loopt vanaf 0
```

# Bugs en errors -> Scope

```
function setup() {  
  let variabele = 10;  
}  
function draw() { Kan niet bij de variabele omdat hij in een andere functie gedefinieerd is  
  console.log(variabele);  
}
```

# Bugs en errors -> Scope

```
function setup() {  
  let variabele = 10;  
}  
function draw() {  
  let variabele = 20;  
  console.log(variabele);  
}
```

Hier komt 20 uit, hij weet van het bestaan van de '10' niets af



# Bugs en errors -> Volgorde

```
console.log(variabele) // ERROR ReferenceError, variabele bestaat niet
let variabele;
function setup() {
  console.log(variabele) // undefined Variabele is nog leeg, wordt pas later gevuld
}
function draw() {
  variabele = 10;
  console.log(variabele) // 10
}
```

# Bugs en errors -> Missende functies

```
<script src="sketch.js"></script> <!-- Connect is not defined -->  
<script src="../oscLib.js"></script>
```

sketch.js heeft functies nodig die zijn gedefinieerd in oscLib.js  
hierom moet oscLib.js vóór sketch.js staan

# Vinden van bugs

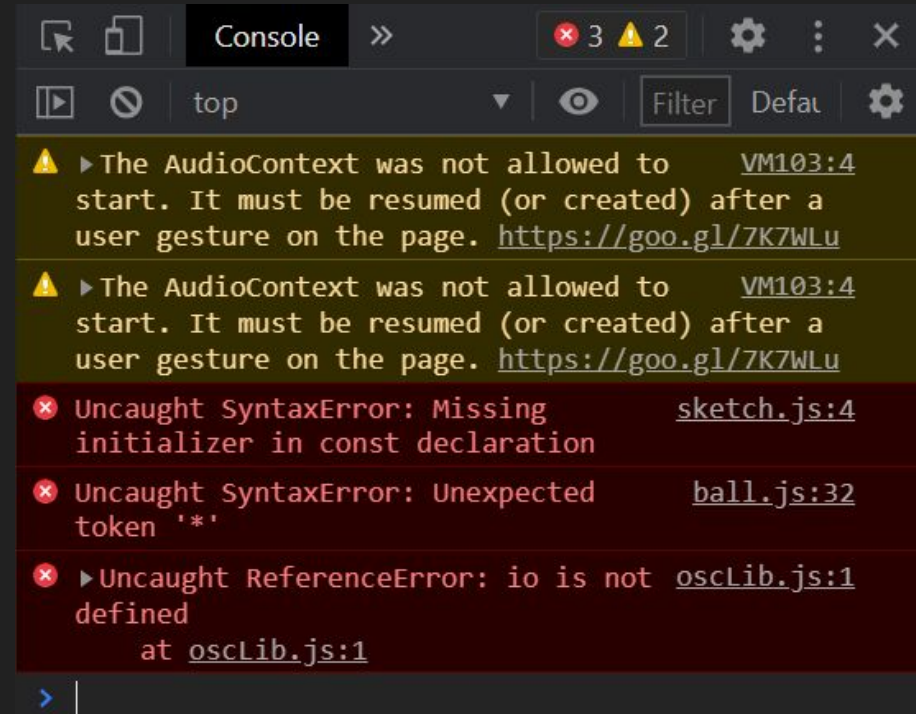
- Error tracer
- De error message
- Regel voor regel
- Opsplitsen
- Print debugging
- Rubber duck debugging
- Peer review

Werkt ook op problemen buiten code  
systematisch nadenken!



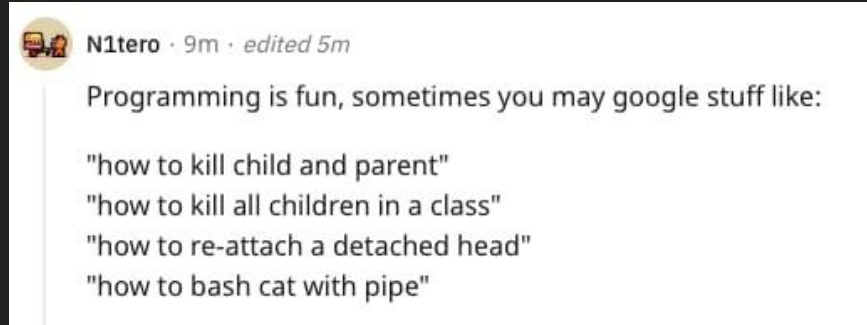
# Vinden van bugs -> Error tracer

- Terminal / F12 / Inspector
- Regelnummer
- Foutcode
- Beschrijving van error



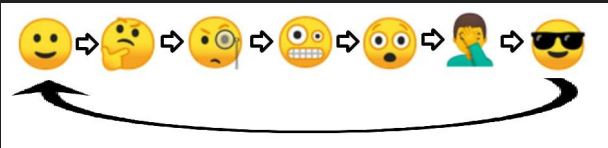
# Vinden van bugs -> De error message

- Niet de hele error message
- Geen pad naar een file
- Zoeken:
  - o CSD syllabus
  - o P5.js reference
  - o Google



# Vinden van bugs -> Regel voor regel

- Alles lezen van begin tot eind tot je eindelijk een keer de fout vind...
- Grote files niet te doen
- Bij kleine stukken code bruikbaar
- Blindstaren...



```
File Edit Selection View ... database.php - AduiwareFrame...
database.php x
public > api > lib > db > database.php > tableExists
1059 /...
1060 * Initialize the database connection.
1061 *
1062 * @return void
1063 */
1064 function setup() {
1065     global $dbType;
1066     if ($dbType == "sqlite") {
1067         return \Database\SQLite\setup();
1068     } else if ($dbType == "mysql") {
1069         return \Database\MySQL\setup();
1070     }
1071 }
1072
1073 /**
1074 * Migrate the database if necessary
1075 *
1076 * @return void
1077 */
1078 function migrate() {
1079     global $dbType;
1080     if ($dbType == "sqlite") {
1081         return \Database\SQLite\migrate();
1082     } else if ($dbType == "mysql") {
1083         return \Database\MySQL\migrate();
1084     }
1085 }
1086
1087 /**
1088 * Check if a table exists
1089 *
1090 * @param string $tableName
1091 * @return void
1092 */
1093 function tableExists($tableName) {
1094     global $dbType;
1095     if ($dbType == "sqlite") {
1096         return \Database\SQLite\tableExists($tableName);
1097     } else if ($dbType == "mysql") {
1098         return \Database\MySQL\tableExists($tableName);
1099     }
1100 }
```

# Vinden van bugs -> Opsplitsen

- Kleine stukjes uitvoeren en testen of het probleem zich daar voordoet

```
draw() {  
  for (let step = 0; step < this.stepsAmount;  
    // const state = this.states[step];  
    // if (running && step == this.currentStep) {  
    //   if (state) fill('#f00')  
    //   else fill('#faa')  
    // }  
    // else {  
    //   if (state) fill('#aaa')  
    //   else fill('#fff')  
    // }  
    | const posX = this.posX + (this.rectSize +  
    // rect(posX, this.posY, this.rectSize, t  
    // this.play(step)  
    }  
  }  
}
```

You, seconds ago • Uncommitted changes

# Vinden van bugs -> Print debugging

- Overall een print tussen zetten zodat je weet wat er gebeurt
- `console.log(variabele);`
- `console.log("var", variabele);`

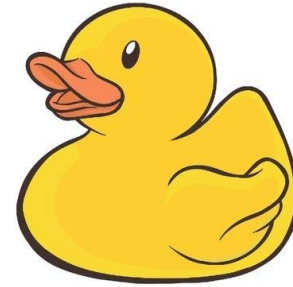
```
draw() {  
  for (let step = 0; step < this.stepsAmount; step++) {  
    const state = this.states[step];  
    console.log("state", state)  
    console.log("running", running)  
    console.log("step", step)  
    console.log("this.currentStep", this.currentStep)  
    if (running && step == this.currentStep) {  
      if (state) fill('#f00')  
      else fill('#faa')  
    }  
    else {  
      if (state) fill('#aaa')  
      else fill('#fff')  
    }  
    const posX = this.posX + (this.rectSize + this.padding) * step  
    console.log("rect", posX, this.posY, this.rectSize, this.rectSize)  
    rect(posX, this.posY, this.rectSize, this.rectSize)  
    this.play(step)  
  }  
}
```

You, seconds ago • Uncommitted changes



# Vinden van bugs -> Rubber duck debugging

- Aan een ander uitleggen wat het probleem is



**HAVE  
YOU  
TRIED**

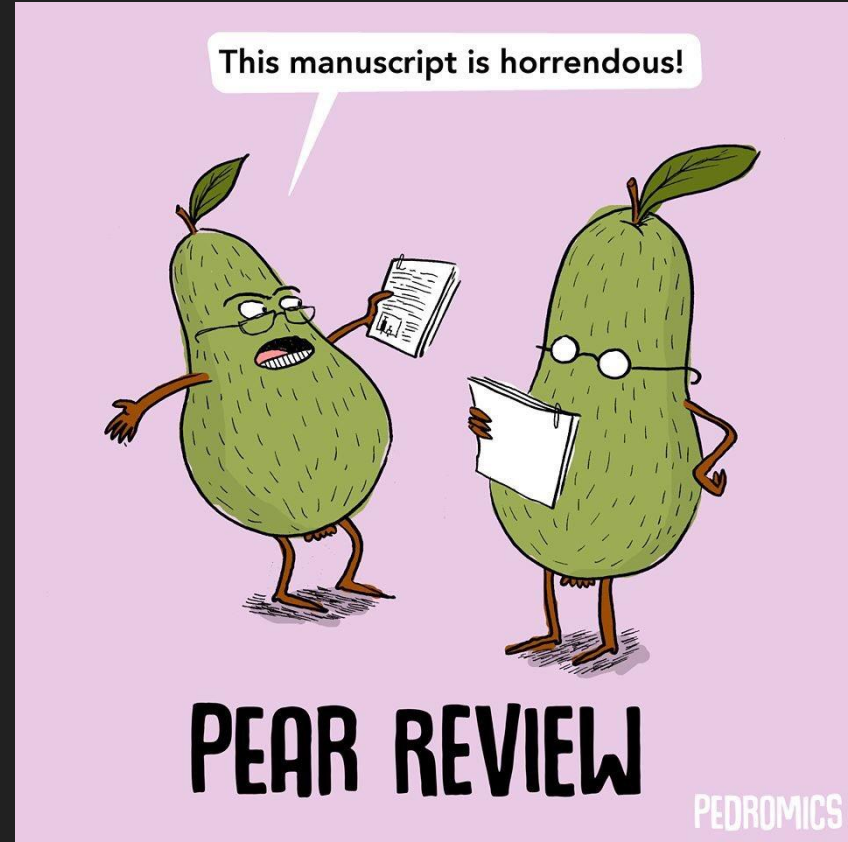
**EXPLAINING**

==== **IT TO THE** ====

**RUBBER DUCK?**

# Vinden van bugs -> Peer review

- Overheen lezen
- Fout 'normaal' vinden



# Voorkomen

- Taal of syntax beter kennen
- Variabel namen
- Opsplitsen
- Overzicht door middel van indents
- Comments

```
Why do developers  
like dark mode?
```

```
Because light attracts  
bugs
```

# Voorkomen -> Taal of syntax beter kennen

Iedere taal is anders:

```
count(array)
```

```
sizeof(array)
```

```
len(array)
```

```
array.length
```

```
array.length()
```

# Voorkomen -> Variabel namen

- Naam net zo veel betekenis als de waarde in de variabele
- Niet te lang, niet te kort
- Gebruik één hoofdtaal (Engels)
- Gebruik geen emojis

```
let r = 10
```

```
let rIsDeRadiusVanDeEersteBal = 10
```

# Voorkomen -> Variabel namen

## Standaard naamgeving regels

- camelCase (variabelen)
  - Alle woorden een hoofdletter behalve de eerste
- PascalCase (classes)
  - Alle woorden hebben een hoofdletter
- snake\_case (variabelen)
  - spaties vervangen door een underscore ( \_ )
- kebab-case (css)
  - Alles kleine letters, spaties vervangen door een min

# Voorkomen -> Opsplitsen

In plaats van:

```
if ( (mouseX > this.posX && mouseX < this.posX + this.rectSize) && (mouseY > this.posY && mouseY < this.posY + this.rectSize)) {  
  console.log("De muis is ingedrukt op " + mouseX + ", " + mouseY);  
}
```

Doe je dit:

```
let insideXBounds = (mouseX > this.posX) && (mouseX < this.posX + this.rectSize);  
let insideYBounds = (mouseY > this.posY) && (mouseY < this.posY + this.rectSize);  
if (insideXBounds && insideYBounds) {  
  console.log("De muis is ingedrukt op " + mouseX + ", " + mouseY);  
}
```

# Voorkomen -> Overzicht

```
draw() {  
  for (let step = 0; step < this.stepsAmount; step++) {  
    const state = this.states[step];  
    if (running && step == this.currentStep) {  
      if (state) {  
        fill('#f00');  
      }  
      else {  
        fill('#faa');  
      }  
    }  
    else {  
      if (state) {  
        fill('#aaa');  
      }  
      else {  
        fill('#fff');  
      }  
    }  
    const posX = this.posX + (this.rectSize + this.padding) * step;  
    rect(posX, this.posY, this.rectSize, this.rectSize);  
    this.play(step);  
  }  
}
```

```
draw() {  
  for (let step = 0; step < this.stepsAmount; step++) {  
    const state = this.states[step];  
    if (running && step == this.currentStep) {  
      if (state) {  
        fill('#f00');  
      }  
      else {  
        fill('#faa');  
      }  
    }  
    else {  
      if (state) {  
        fill('#aaa');  
      }  
      else {  
        fill('#fff');  
      }  
    }  
    const posX = this.posX + (this.rectSize + this.padding) * step;  
    rect(posX, this.posY, this.rectSize, this.rectSize);  
    this.play(step);  
  }  
}
```



# Voorkomen -> Comments

- Beschrijf een regel of een stuk code zodat een ander het zonder jou kan begrijpen
- Over een jaar ben jij zelf 'een ander'

Oefenen

# Oefenen

- Download de opdracht op de website
- Los deze bugs op en noteer:
  - Wat was de bug
  - Waar vond je de bug
  - Hoe heb je de bug gevonden
  - Hoe heb je de bug opgelost
- Aan het einde komen de antwoorden met uitleg online

# Sysbas1A

Debuggen