Intro/intro.md

Welcome to my presentation

Intro/intro.md

Welcome to my presentation

Intro/problem.md

Problem: too difficult for non-english speaking arabs to learn programming

Intro/problem.md

Problem: too difficult for non-english speaking arabs to learn programming

Solution: make an arabic programming language :)

Intro/title.md

عرب Script

< By Khalid Alghunaim >

Special Thanks/Lee.jpeg

❤️

Special Thanks/omar.jpeg

Special Thanks/dr_toal.jpeg

< Syntax Designer: Omar Abdulatif >

Special Thanks/omar.jpeg

Special Thanks/dr_toal.jpeg

< Mentor: Ray Toal >

Key_Features {

- Dynamically Typed
- Static Scope
- Types:
  - Integer
  - Float
  - Boolean
  - String
  - Array
  - Object
  - Undefined/Null
- It's in Arabic!

}

Key_Features {

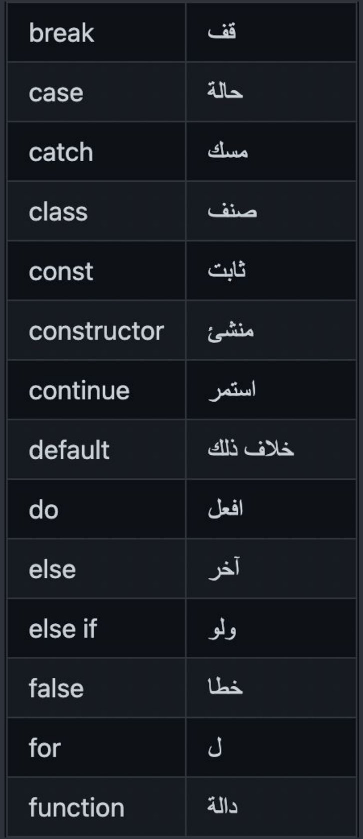
- Multiple Declarations on the same line
- Ternary
- Try Catch
- Ifs
- Switchs
- For/ For of
- While/ Do While
- Functions
- Classes
- Even built in functions like console.log and typeof!

}

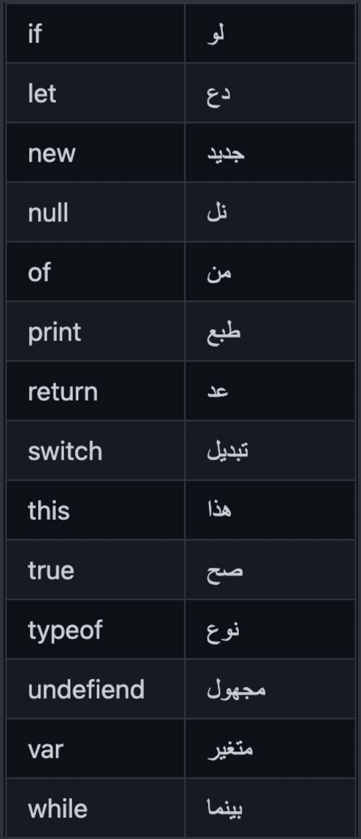
My Process

1) Syntax/Parser

Ohm was used to parse the expression grammars



break	قف
case	حالة
catch	مسك
class	صنف
const	ثابت
constructor	منشئ
continue	استمر
default	خلاف ذلك
do	افعل
else	آخر
else if	ولو
false	خطا
for	ل
function	دالة



if	لو
let	دع
new	جديد
null	نل
of	من
print	طبع
return	عد
switch	تبديل
this	هذا
true	صح
typeof	نوع
undefiend	مجهول
var	متغير
while	بينما

My Process

1) Syntax/Parser

Ohm was used to parse the expression grammars

2) Analyzer

After creating all the ast nodes, we need to analyze them!

- Type checking
- Is Iterable?
- Is a Constant?
- Are we inside a function, class or loop?
- Are all keys of an object distinct?

My Process

1) Syntax/Parser

Ohm was used to parse the expression grammars

Examples:
خالد → Khaled
ن → var_n

2) Analyzer

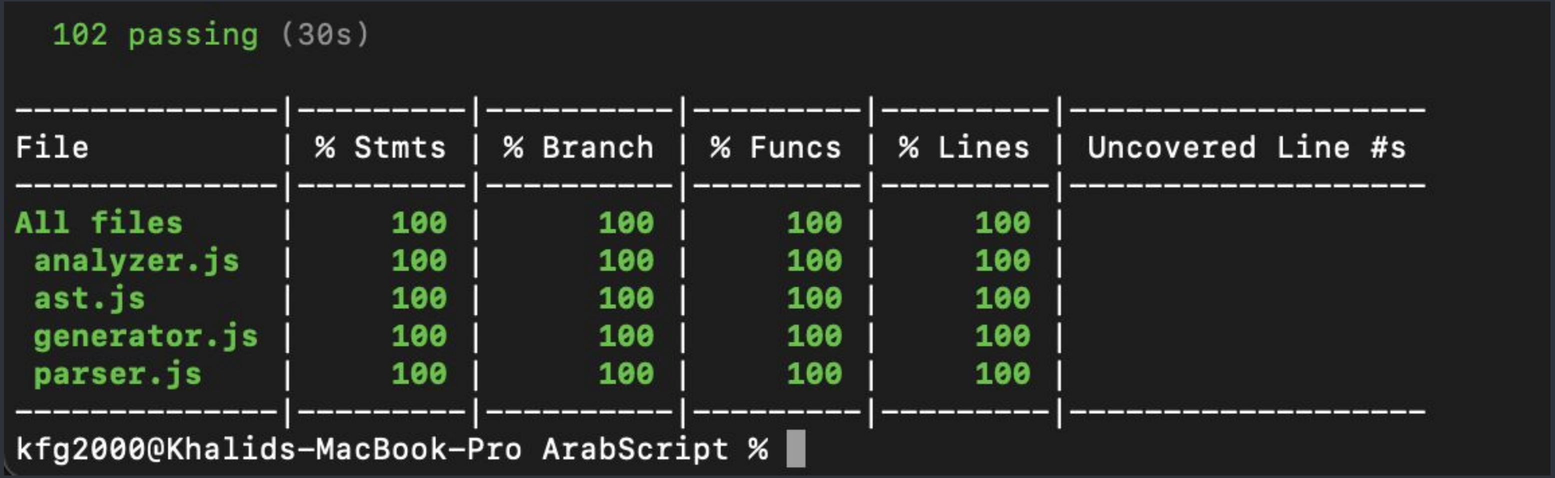
After creating all the ast nodes, we need to analyze them!

3) Generator

Lastly, we generate javascript code. Using the NLP Translation API, we can translate variable names to english!

My Process

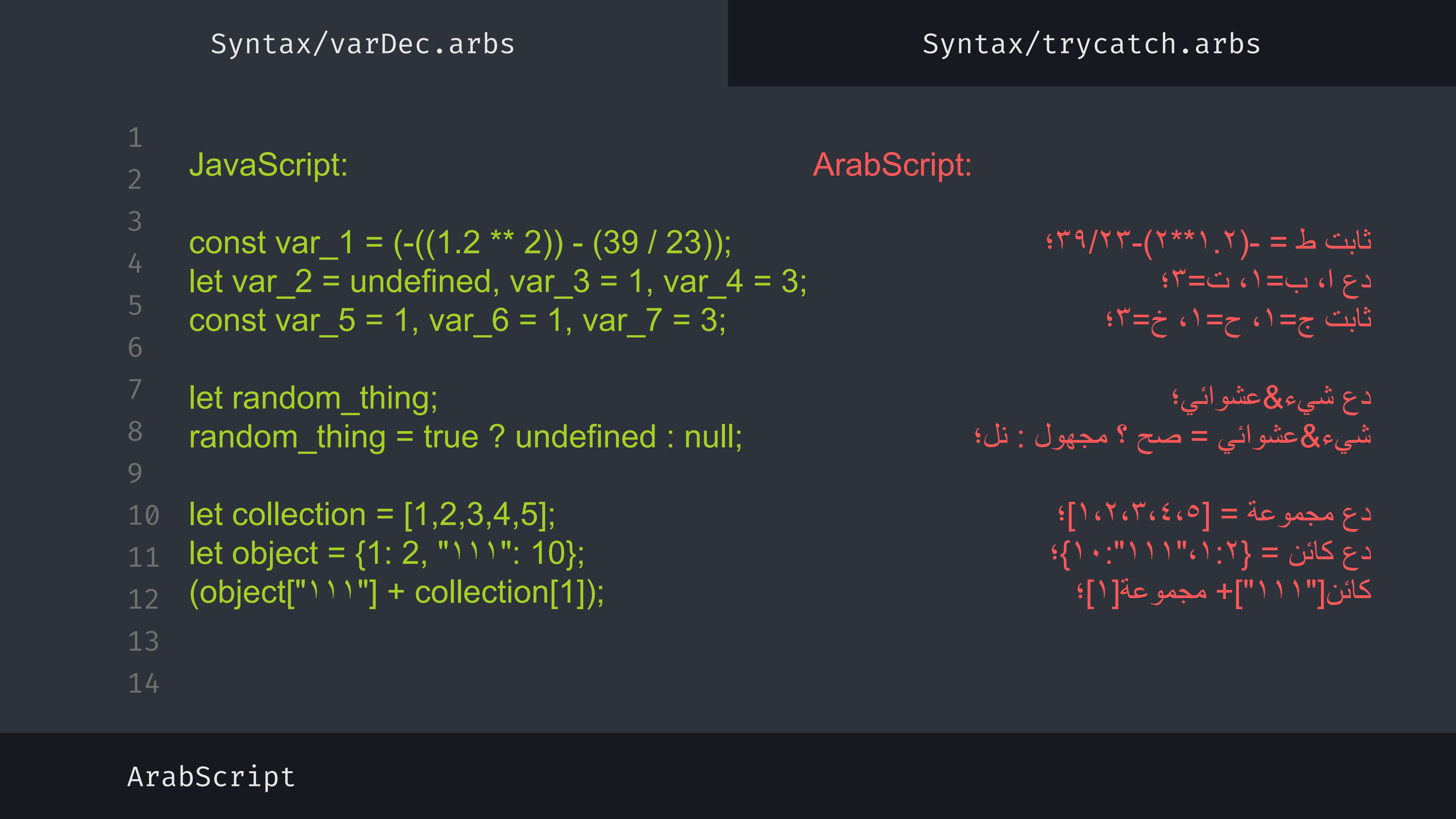
4) Test



```
  102 passing (30s)

--------------|----------|-----------|----------|----------|--------------------
File          | % Stmts  | % Branch  | % Funcs  | % Lines  | Uncovered Line #s
--------------|----------|-----------|----------|----------|--------------------
All files     |      100 |       100 |      100 |      100 |
 analyzer.js  |      100 |       100 |      100 |      100 |
 ast.js       |      100 |       100 |      100 |      100 |
 generator.js |      100 |       100 |      100 |      100 |
 parser.js    |      100 |       100 |      100 |      100 |
--------------|----------|-----------|----------|----------|--------------------
kfg2000@Khalids-MacBook-Pro ArabScript %
```

Special Thanks/Lee.jpeg



```
JavaScript:

const var_1 = (-((1.2 ** 2)) - (39 / 23));
let var_2 = undefined, var_3 = 1, var_4 = 3;
const var_5 = 1, var_6 = 1, var_7 = 3;

let random_thing;
random_thing = true ? undefined : null;

let collection = [1,2,3,4,5];
let object = {1: 2, "١١١": 10};
(object["١١١"] + collection[1]);
```

```
ArabScript:

ثابت ط = -(١.٢**٢)-٣٩/٢٣؛
دع ا، ب=١، ت=٣؛
ثابت ج=١، ح=١، خ=٣؛

دع شيء&عشوائي؛
شيء&عشوائي = صح ؟ مجهول : نل؛

دع مجموعة = [١،٢،٣،٤،٥]؛
دع كائن = {١:٢،"١١١":١٠}؛
كائن["١١١"]+ مجموعة[١]؛
```

Syntax/varDec.arbs

Syntax/trycatch.arbs

JavaScript:

```
let var_1 = {};
try {
        var_1.feature.feature;
} catch(mistake) {
        console.log(mistake);
}
```

ArabScript:

```
دع ا = {}؛
{ ا.خاصية.خاصية؛ } مسك(غلطة) { طبع(غلطة)؛ }
```

Syntax/if.arbs   Syntax/switch.arbs

JavaScript:

```
let noun = "خالد";
if ((noun === "خالد")) {
	console.log("What a lad");
} else if (((noun === "نور") || (noun === "بدر"))) {
	console.log("Meh");
} else {
	console.log("Not a sibling");
}
```

ArabScript:

```
دع اسم = "خالد"؛
لو(اسم == "خالد"){
	طبع("What a lad")؛
}ولو(اسم == "نور" || اسم == "بدر"){
	طبع("Meh")؛
}آخر{
	طبع("Not a sibling")؛
}
```

Syntax/if.arbs   Syntax/switch.arbs

JavaScript:

```
let noun = "خالد";
switch(noun) {
    case "خالد":
        console.log("Lead Programmer");
        break;
    case "عمر":
        console.log("Syntax Designer");
        break;
    case "راي تول":
        console.log("Mentor");
        break;
    default: console.log("Morale support");
}
```

ArabScript:

```
دع اسم = "خالد"؛
تبديل(اسم){
    حالة "خالد":
        طبع("Lead Programmer")؛
        قف؛
    حالة "عمر":
        طبع("Syntax Designer")؛
        قف؛
    حالة "راي تول":
        طبع("Mentor")؛
        قف؛
    خلاف ذلك: طبع("Morale support")؛
}
```

ArabScript

Syntax/for.arbs   Syntax/while.arbs

JavaScript:

```
for (let number = 1; (number <= 5); number++) {
        console.log(number);
}

let numbers = [1,2,3,4,5];
for (const number of numbers) {
        console.log(number);
}
```

ArabScript:

```
ل(دع رقم = ١ ؛ رقم<=٥؛رقم++){
      طبع(رقم)؛
}

دع ارقام = [١،٢،٣،٤،٥]؛
ل(رقم من ارقام){
      طبع(رقم)؛
}
```

Syntax/for.arbs Syntax/while.arbs

JavaScript:

```
function countdown(number) {
        console.log("Begin countdown!");
        while ((number > 0)) {
                console.log(number);
                number--;
        }
        console.log("Blast off!");
}
do {
        countdown(10);
} while (false);
```

ArabScript:

```
دالة العد&التنازلي(عدد){
        طبع("Begin countdown!")؛
        بينما(عدد > ٠){
                طبع(عدد)؛
                عدد--؛
        }
        طبع("Blast off!")؛
}
افعل{
        العد&التنازلي(١٠)؛
}بينما(خطا)؛
```

Syntax/functions.arbs   Syntax/classes.arbs

JavaScript:

```
function fibonacci(var_1) {
        if ((var_1 <= 1)) {
                return var_1;
        }
        return (fibonacci((var_1 - 1)) +
        fibonacci((var_1 - 2)));
}
```

ArabScript:

```
دالة فيبوناتشي(ن){
 لو(ن <= ١){
  عد ن؛
 }
 عد فيبوناتشي(ن-١) + فيبوناتشي(ن-٢)؛
}
```

JavaScript:

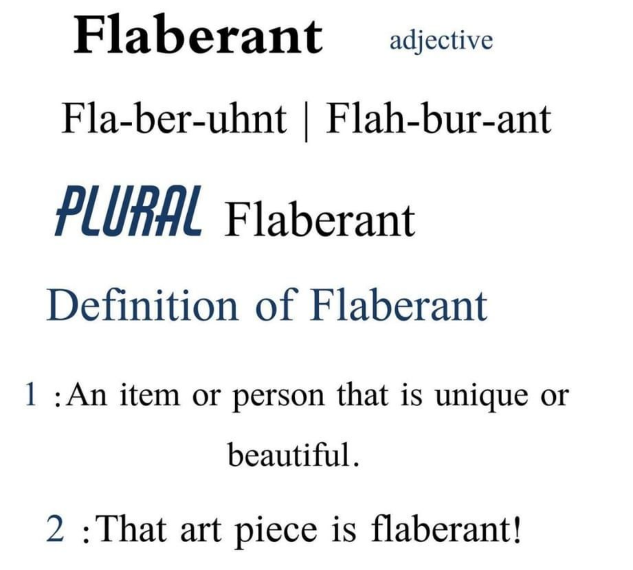
```
class dog{
        constructor (noun, age){
                this.noun = noun;
                this.age = age;
        }
        function bless() {
                console.log("ووف");
        }
}

let khaled = new dog("7 ,"خالد");
khaled.noun = "احمد";
khaled.bless();
```

ArabScript:

```
صنف كلب {
    منشئ(اسم،عمر){
        اسم هذا = اسم؛
        عمر هذا = عمر؛
    }
    دالة بارك(){
        طبع("ووف")؛
    }
}
دع خالد = كلب("خالد"،٧) جديد؛
خالد.اسم = "احمد"؛
خالد. بارك()؛
```

languageDescription.jpeg



Flaberant adjective

Fla-ber-uhnt | Flah-bur-ant

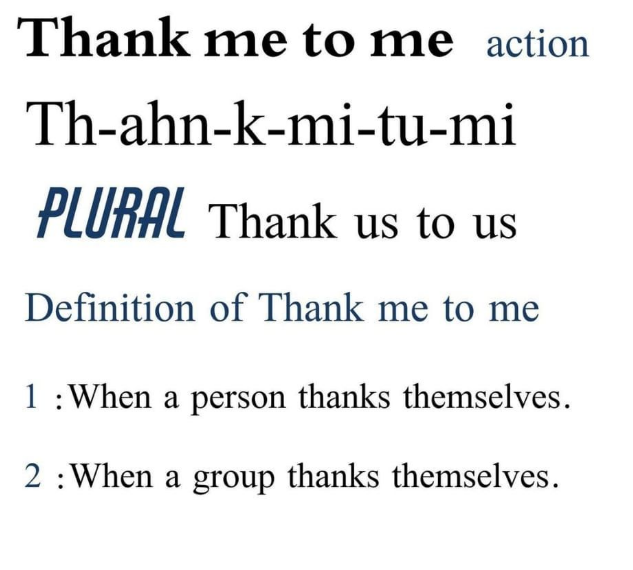
PLURAL Flaberant

Definition of Flaberant

1 :An item or person that is unique or beautiful.

2 :That art piece is flaberant!

Fatma-Webster®



Thank me to me action

Th-ahn-k-mi-tu-mi

PLURAL Thank us to us

Definition of Thank me to me

1 :When a person thanks themselves.

2 :When a group thanks themselves.

end.md

Thanks; {

‘Do you have any questions?’

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, and infographics & images by **Freepik**

}

end.md

Thanks; {

‘Do you have any questions?’

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, and infographics & images by **Freepik**

}