

## Auto speed “parser tree”

### **Start**

$\langle \text{Program} \rangle \Rightarrow \langle \text{ImportSection} \rangle | \langle \text{NamespaceSection} \rangle | \langle \text{FunctionList} \rangle$

### **Import & Namespace**

$\langle \text{ImportSection} \rangle \Rightarrow \langle \text{ImportDecl} \rangle | \langle \text{ImportSection} \rangle | \epsilon$

$\langle \text{ImportDecl} \rangle \Rightarrow \text{"\#oil" " <Identifier> "}$

$\langle \text{NamespaceSection} \rangle \Rightarrow \text{"key" <Identifier>}$

### **Function Definitions**

$\langle \text{FunctionList} \rangle \Rightarrow \langle \text{Function} \rangle | \langle \text{FunctionList} \rangle | \epsilon$

$\langle \text{Function} \rangle \Rightarrow \text{"engine" <Identifier> "(" <ParamList> ")" <Block>}$   
 $\quad | \text{"ignite" "(" <ParamList> ")" <Block>}$

$\langle \text{ParamList} \rangle \Rightarrow \langle \text{Param} \rangle \{ \text{" , " } \langle \text{Param} \rangle \} | \epsilon$

$\langle \text{Param} \rangle \Rightarrow \langle \text{Type} \rangle | \langle \text{Identifier} \rangle$

### **Blocks & Statements**

$\langle \text{Block} \rangle \Rightarrow \text{"{" <StatementList> "}"}$

$\langle \text{StatementList} \rangle \Rightarrow \langle \text{Statement} \rangle | \langle \text{StatementList} \rangle | \epsilon$

$\langle \text{Statement} \rangle \Rightarrow \langle \text{DeclarationStmt} \rangle$   
 $\quad | \langle \text{AssignmentStmt} \rangle$

| <ConditionalStmt>

| <LoopStmt>

| <OutputStmt>

| <InputStmt>

| <FunctionCallStmt>

| <ReturnStmt>

### **Variable Declaration & Assignment**

<DeclarationStmt> => <Type>| <Identifier> "=" <Expression> ";"

<AssignmentStmt> => <Identifier> "=" <Expression> ";"

<ConditionalStmt> => "track" "(" <Expression> ")" <Block> [ "pitstop" <Block> ]

<LoopStmt> ::= "looplap" "(" <Expression> ")" <Block>

### **I/O & Function Calls**

<OutputStmt> => "announce" <Expression> ";"

<InputStmt> => "listen" <Identifier> ";"

<FunctionCallStmt> => <Identifier> "(" <ArgList> ")" ";"

<ReturnStmt> => "finishline" <Expression> ";"

### **Arguments**

<ArgList> => <Expression> { "," <Expression> } | ε

### **Expressions**

<Expression> => <Term> { ("+" | "-" | "==" | "!=" | "<" | "<=" | ">" | ">=") <Term> }

<Term> => <Factor> { ("\*" | "/" | "%") <Factor> }

<Factor> => <Number>

| <String>

| <Identifier>

| "(" <Expression> ")"

| <FunctionCallExpr>

| <BooleanLiteral>

<FunctionCallExpr> => <Identifier> "(" <ArgList> ")"

<BooleanLiteral> => "true" | "false"

### **Data Types**

<Type> => "gear" | "track" | "exhaust" | "turbo" | "flag"

### **Tokens / Lexical Units**

<Identifier> => letter { letter | digit | "\_" }

<Number> => digit { digit } [ "." digit { digit } ]

<String> => '"' { character } '"'