

skyguide

ENSG
Géomatique

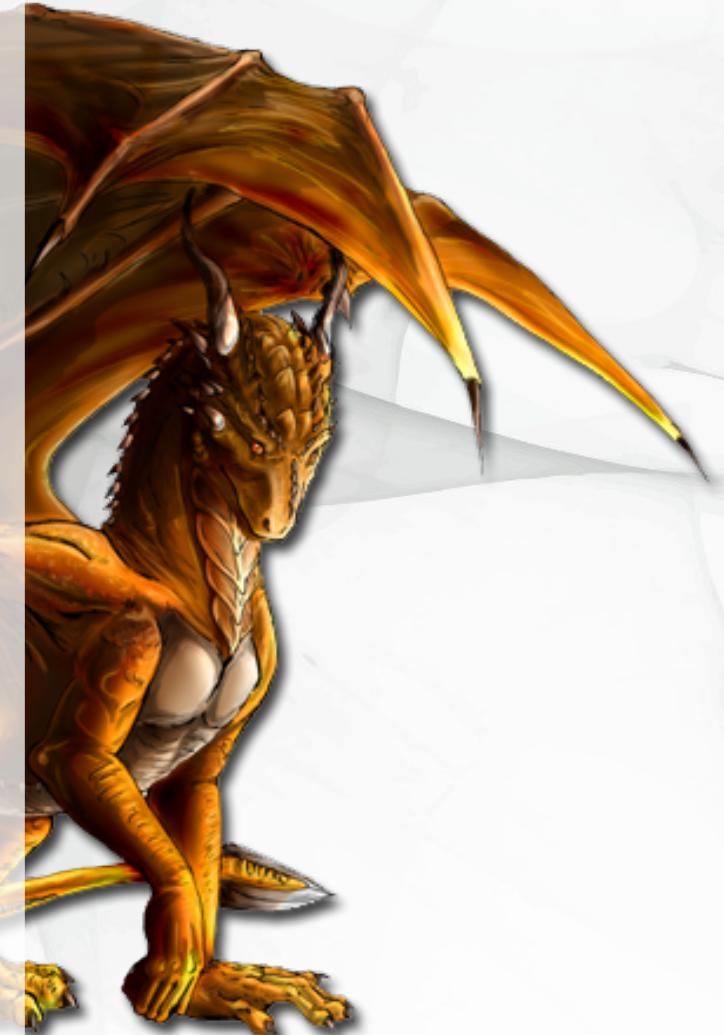
Compilation & langages

Présenté par Yann Caron
skyguide

ENSG Géomatique

Plan général du cours

- 00 - Introduction
- 01 - Compilation
- 02 - CodeGen
- 03 - Principes et paradigmes
- 04 - Théorie des langages



Présentation

Yann Caron

skyguide – SCADA – Java – C# - C++

Ingénieur EI-CNAM

Projet Algoid www.algoid.net & Wikipédia

Projet JinyParser

<https://github.com/YannCaron/JinyParser>

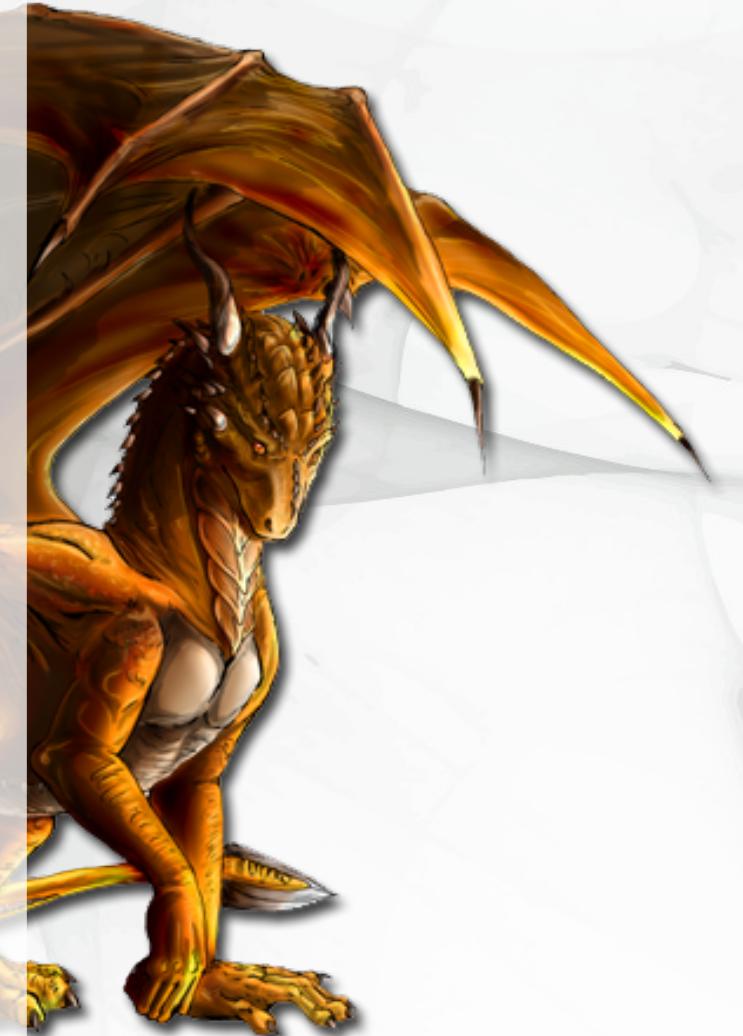
Plan du cours

Définitions

Historique des langages

Références

Outils



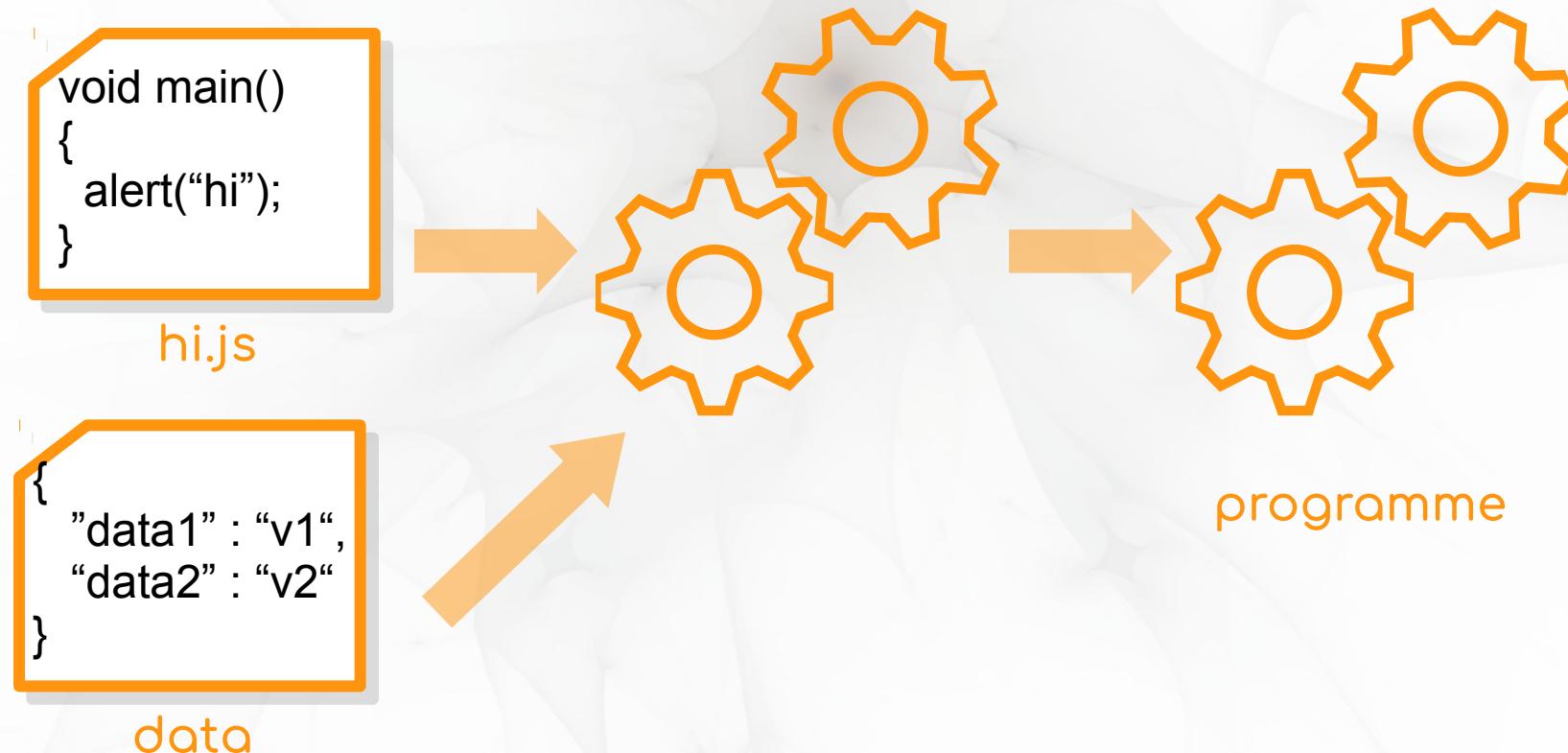
Compilateur - définition

Programme qui transforme un code source (langage source) dans un langage cible
Ex : binaire ou C++ vers du C



Interpréteur - définition

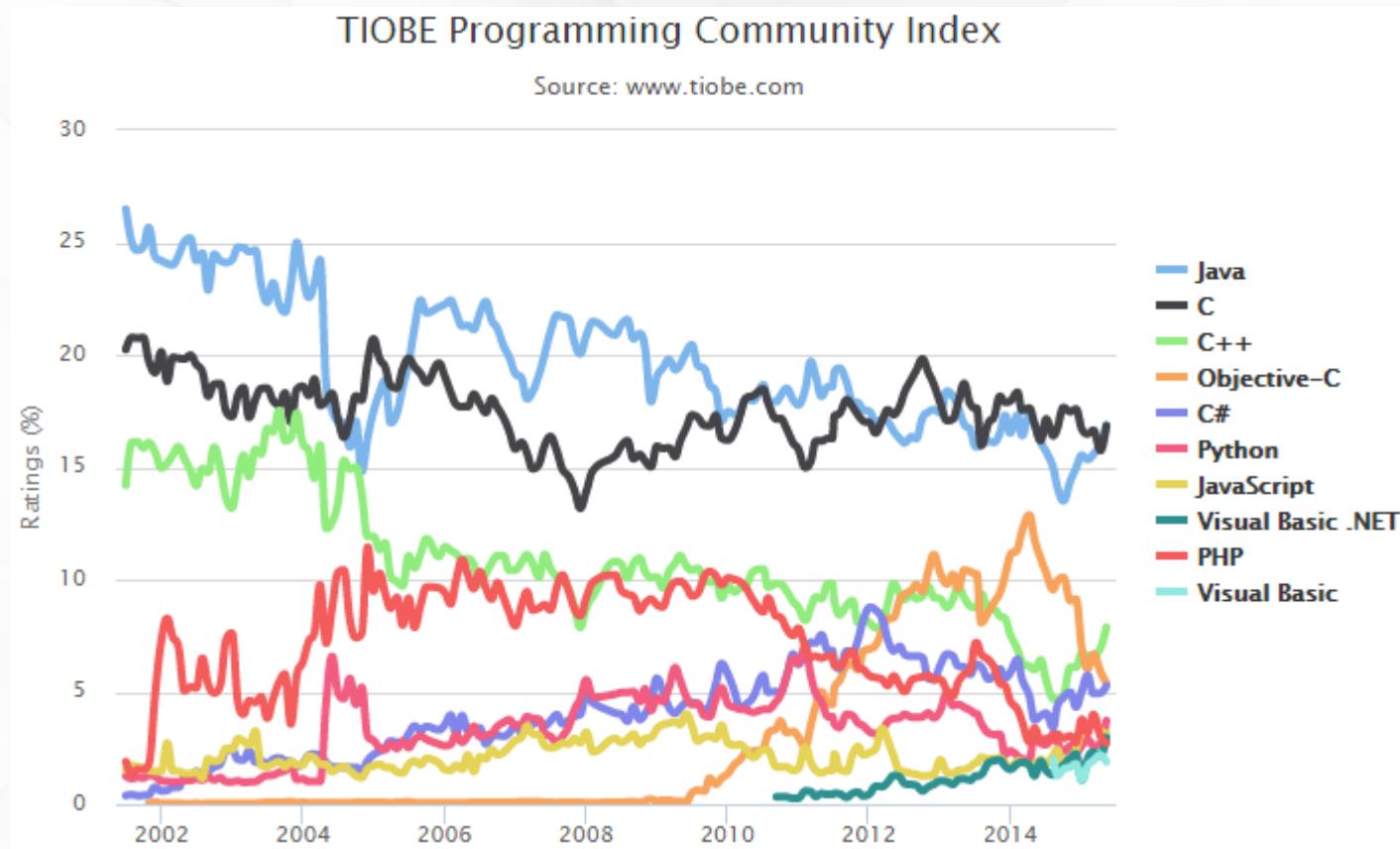
Programme qui transforme un code source (langage source) en un résultat



Philosophie

Quel est le meilleur langage ?

Le plus utilisé ?



Philosophie

- ✓ pas de meilleurs langages que d'autres
- ✓ des langages pour des besoins différents
- ✓ script
 - ✓ pour : rapide, simple efficace
 - ✓ contre : librairies, bugs difficiles à identifier
- ✓ compilés
 - ✓ pour : langages sûr, correction à la compilation
 - ✓ contre : verbeux, complexes, rigides

Par domaines

Domaine	Fonctionnalités	Langage
Scientifique	Virgule flottante Tableaux Parallélisme	Fortran
Business	Persistante Génération de rapports Analyse des données	SQL
Programmation système	Contrôle des ressources Temps réel	C / C++

Compilation ? Pourquoi ?

- ✓ Challenge intellectuel
 - ✓ Paradigmes
 - ✓ Algorithmes
 - ✓ Compréhension en profondeur des langages
- ✓ Économiques ex : Oracle vs Microsoft
- ✓ Besoin de configuration avancé ! (exemple projet Stars)
- ✓ Domain Specific Languages

Économiques

- ✓ Atlassian
- ✓ Sonar-source
- ✓ Oracle vs Microsoft

Nouveau mais !

- ✓ Les nouveaux langages créés tendent à ressembler aux anciens
- ✓ Exemple : C++ - Java - C#
- ✓ TypeScript, JavaScript - Algoid
- ✓ Parce que l'apprentissage d'un nouveau langage coûte cher
- ✓ en regard à ce qu'il apporte comme nouveautés

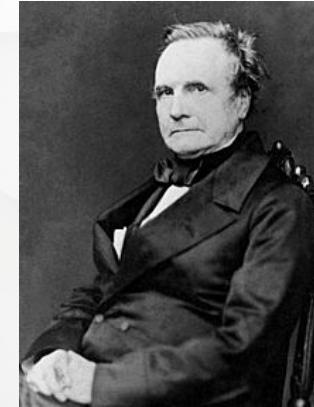
Historiques des langages



Introduction

- ✓ La discipline informatique a pré-existée aux ordinateurs
- ✓ Alonzo Church, Alan Turing, Stephen Kleene et Kurt Gödel participent à cette recherche :
- ✓ Thèse de Church-Turing qui définit un modèle “physique” de calculabilité
- ✓ Ils définissent ce qu'est un algorithme (ensemble fini d'instructions simples)
- ✓ Les Lambda-Calcul et les fonctions récursives

Avant 1940



Charles Babbage Ada Lovelace
1791 - 1871 1815 - 1852

- ✓ Ada Lovelace
- ✓ Augusta Ada king, comtesse de Lovelace
- ✓ Traduit le mémoire du mathématicien Luigi Menabrea sur la machine analytique, la première machine proposé par Charles Babbage
- ✓ Premiers programme et premier algorithme écrit
- ✓ Langage Ada nommé en son honneur

1940

- ✓ Premières machines électromécaniques
 - ✓ La Bombe de Turing en Angleterre
 - ✓ ENIAC, Armée américaine, lié à la bombe nucléaire)
- ✓ Premiers ordinateurs électroniques à lampe (EDSAC / IAS / EDVAC)
- ✓ Principalement programmés en langage machine
- ✓ Ou en assembleur
- ✓ Plankalkül – premier langage de haut niveau

Alan Michael Turing



Alan Michael
Turing
1912 - 1954

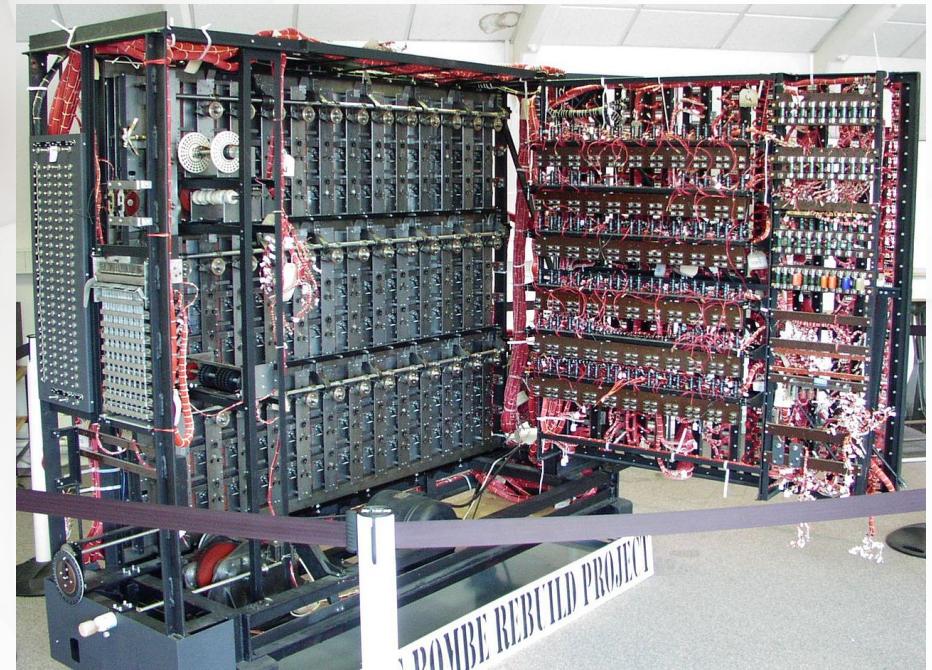
- ✓ Mathématicien Cryptologue
 - ✓ Inventeur du premier ordinateur
 - ✓ Débat sur l'intelligence artificielle : test de Turing
-
- ✓ Inventeur du concept de programmation et de programme
 - ✓ Film à voir : **Imitation Game**

Alan Michael Turing

- ✓ Casse le code Enigma des Nazis durant la 2^{em} guerre mondiale
- ✓ Les historiens estiment qu'il a écourté la guerre de 2 ans
- ✓ Se suicide parce qu'accusé d'homosexualité en mangeant partiellement une pomme empoisonnée au cyanure (logo apple ?)
- ✓ Gracié, puis reconnu héros de guerre 55 ans après son décès

Alan Michael Turing

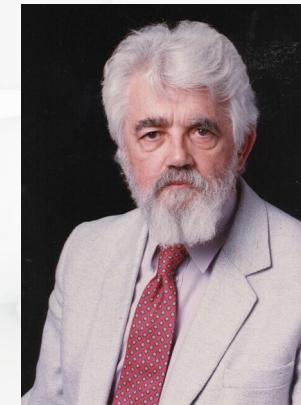
- ✓ La bombe (1938, électromécanique)
- ✓ Testait automatiquement les codes d'Enigma
- ✓ Bletchley Park - RU



1950 - 1960



John Backus
1924 - 2007



John McCarthy
1927 - 2011



Grace Hopper
1906 - 1984

- ✓ FORTRAN (FORmula TRANslator) - John Backus
- ✓ LISP (LISt Processor) spécialisé dans le traitement des listes - John McCarthy
Premier langage fonctionnel basé sur le lambda calcul
- ✓ COBOL (Common Business Oriented Language) – Grace Hopper

Algol



Peter Naurr
1928 - 2016

- ✓ ALGOL (ALGOrithmic Oriented Language) – John Backus et Peter Naurr
- ✓ Apporte la notion de portées (encapsulé) et de bloc de code
- ✓ Création du méta langage BNF (Forme de Backus-Naurr) dont un dérivé (EBNF) sert aujourd’hui encore à la description des langages modernes (du moins la partie context free)

1967 - 1978

- ✓ Mise en place des paradigmes fondamentaux
- ✓ Simula 67 – premier langage Orienté Objet
- ✓ C – premier langage système
- ✓ Smalltalk – premier IDE graphique (Objet et Fonctionnel, UI Fenêtres) – Alan Key
- ✓ Prolog – premier langage Logique
- ✓ ML – pose les bases de la programmation fonctionnelle, typage statique et polymorphique (basé sur Lisp)



Alan Key
1940

1980

- ✓ Normalisation et recherche de performance
- ✓ Ada, Eiffel, Perl
- ✓ C++ - langage système et OO – Bjarne Stroustrup
- ✓ FP – Functional Programming - John Backus



Bjarne Stroustrup
1950

1990



Anders Hejlesberg
1960



James Gosling
1956



Guido van Rossum
1956

- ✓ Les langages se tournent vers internet
- ✓ Python – Guido van Rossum
- ✓ Ruby, Lua, JavaScript, PHP

- ✓ Mais aussi : Haskell
- ✓ Java – James Gosling
- ✓ C# - Hejlesberg (co-inventeur du Pascal)



ADA

Ada Lovelace

Premier langage informatique écrit pour la machine analytique de Charles Babbage

1842

1954

Sert principalement pour le calcul scientifique
Toujours très utilisé

FORTRAN Fortran

Formula Translation



John Backus



LISP

List Processor
Inventé par
John McCarthy

COBOL

Créé par le Short Range Committee dans lequel on trouve Grace Hopper



PASCAL

Niklaus Wirth

Conçu pour l'enseignement
sert à l'écriture de Photoshop et du système Macintosh

1972

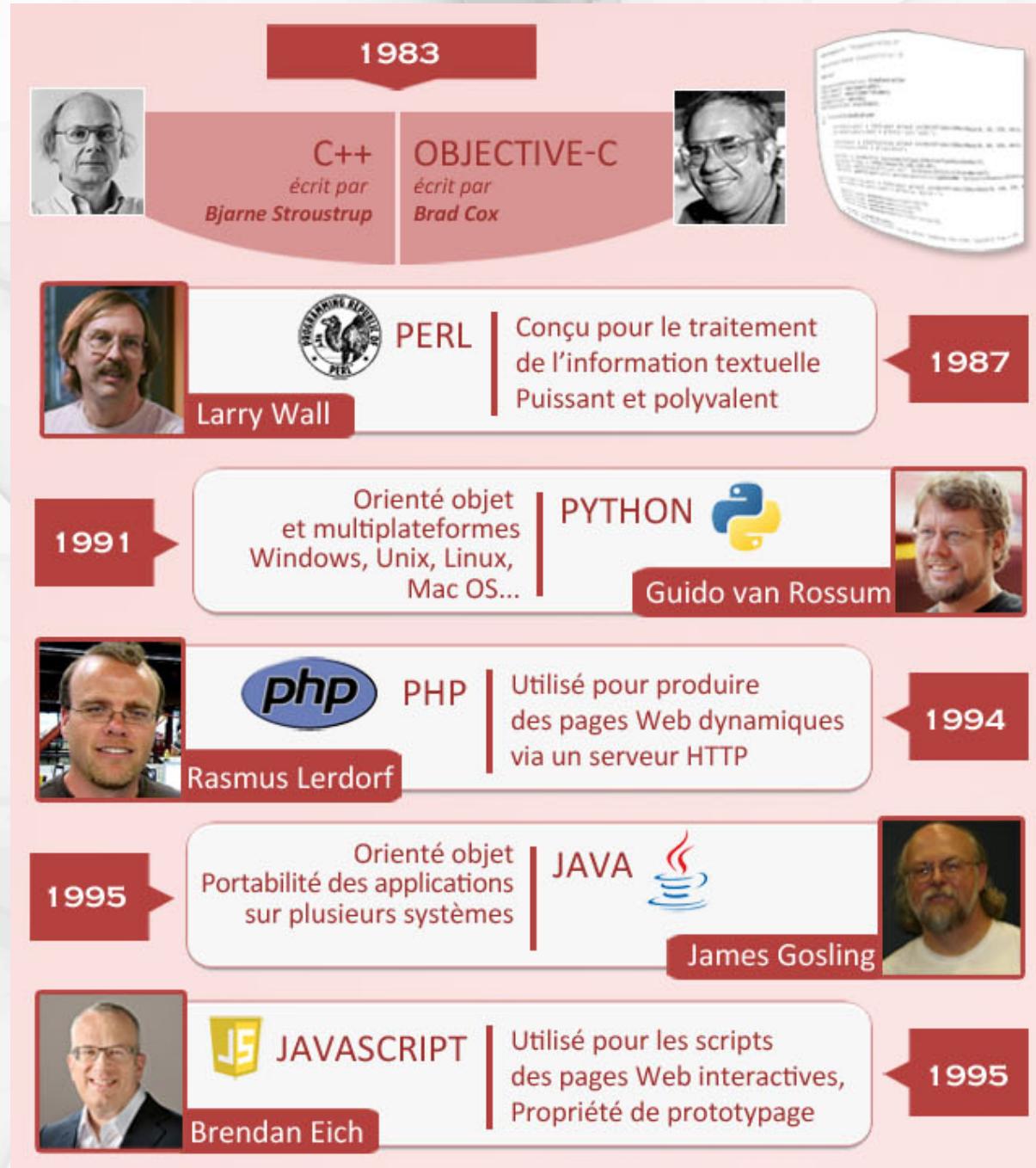
1972

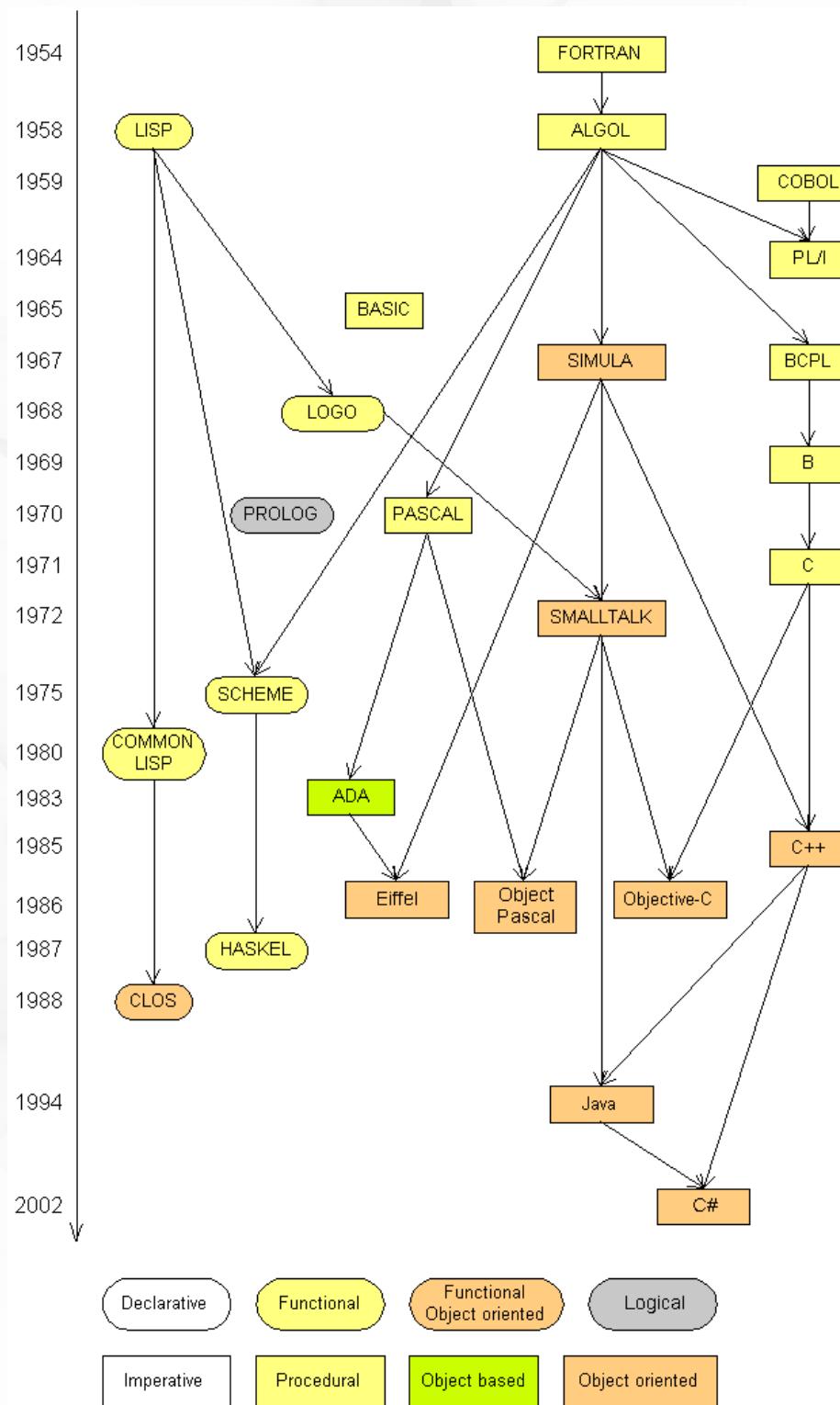
Sert au développement d'UNIX
Toujours très utilisé

LANGAGE C



Dennis Ritchie



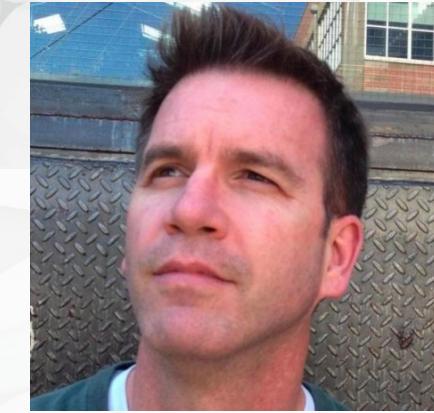
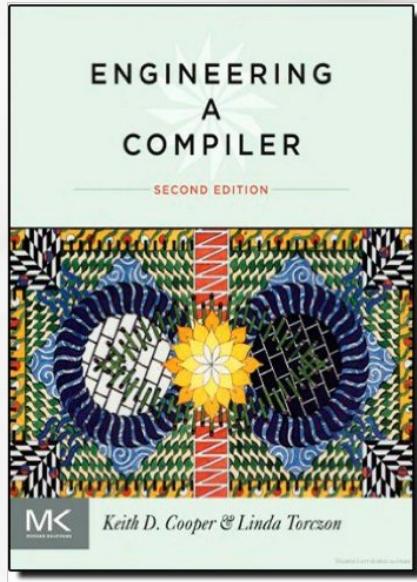
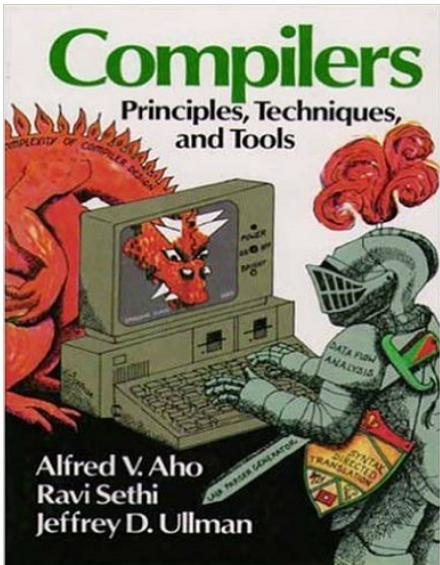


Références

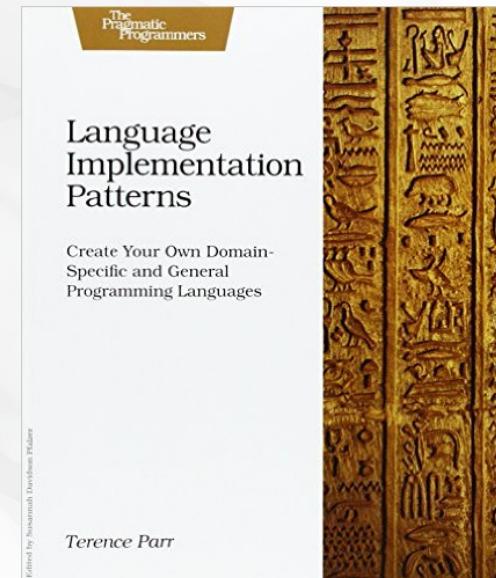


Bibliographie

- ✓ Dragon book
- ✓ Engineering a compiler
- ✓ Terrens Parr



Terrens Parr -1964



MOOCs



Jeff Ullman - 1942



Alex Aiken

- ✓ Stanford Online - Automata - Jeffrey D. Ullman

<http://online.stanford.edu/course/automata-theory-self-paced>

- ✓ Stanford Online - Compilers - Alex Aiken (language COOL)

<https://lagunita.stanford.edu/courses/Engineering/Compilers/Fall2014/about>

MOOCs



Peter Van Roy



Sedgewick - 1946

- ✓ Louvain – Paradigms of Computer Programming (Langage Oz)

<https://www.edx.org/course/paradigms-computer-programming-louvainx-louv1-1x-2>

- ✓ Princeton – Algorithms Part II – Robert Sedgewick

<https://www.coursera.org/learn/algorithms-part2>

Acteurs



Donald Knuth
1938

- ✓ Donald Knuth
 - ✓ Tex → LaTex (Domain specific language)
 - ✓ Parser LR
 - ✓ Grammaire attribués (attributs de l'AST)

Outils



Outils - parsers

- ✓ Pour un langage : GCC, JavaCC, Roslyn
- ✓ Lex / Yacc - Flex / Bison - Génération de code
- ✓ Antlr - BNF (grammaire externe)
- ✓ Irony.net - Surcharge d'opérateurs

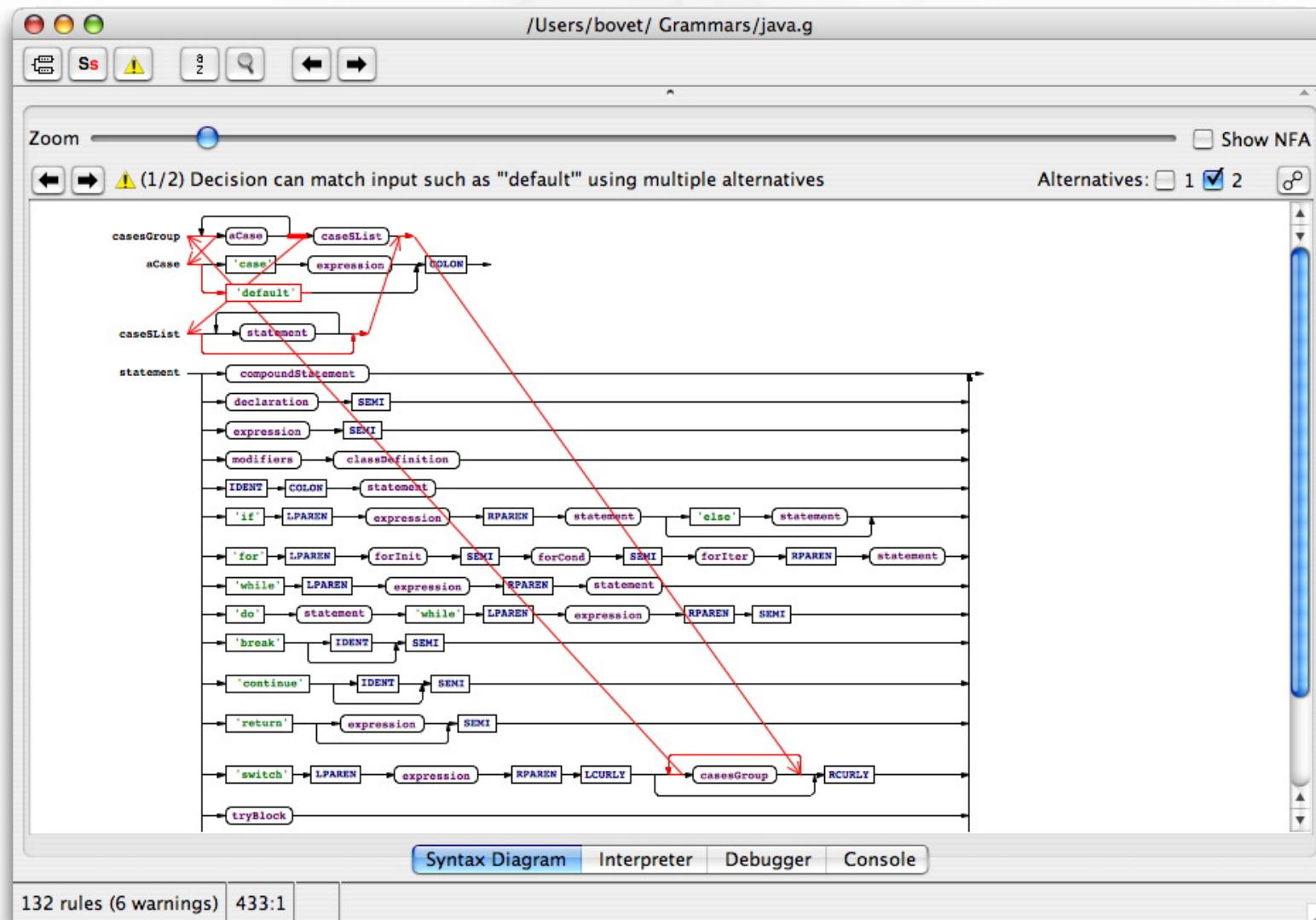
```
ParExpr.Rule = "(" + Expr + ")";
```
- ✓ JParsec - Déclaratif fonctionnel
- ✓ JASI - JinyParser

Outils - compilateurs

- ✓ ASM – Générateur de bytecode JVM et Dalvik – ex : Golo
- ✓ CECIL, ILasm – Générateur d'Intermediate Language .net
- ✓ LLVM – VM C / C++
- ✓ Neko VM (bytecode de haut niveau)



Outils – AntlrWorks



Outils – Railroad

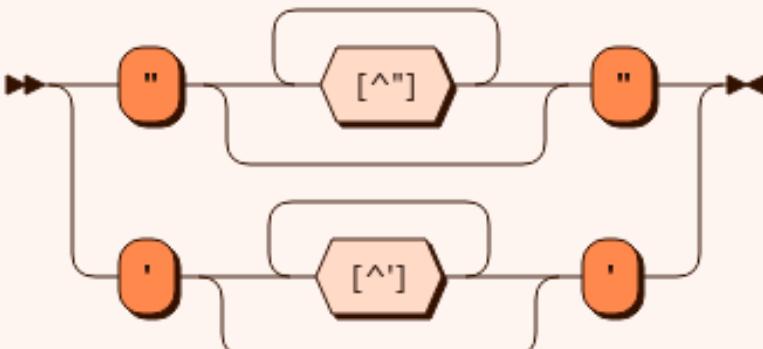
Written by Gunther Rademacher
grd@gmx.de

Railroad Diagram Generator
v1.47.1494 Apr 15, 2017

SAXON
XQUER

Welcome **Get Grammar** **Edit Grammar** **View Diagram** **Options**

StringLiteral:



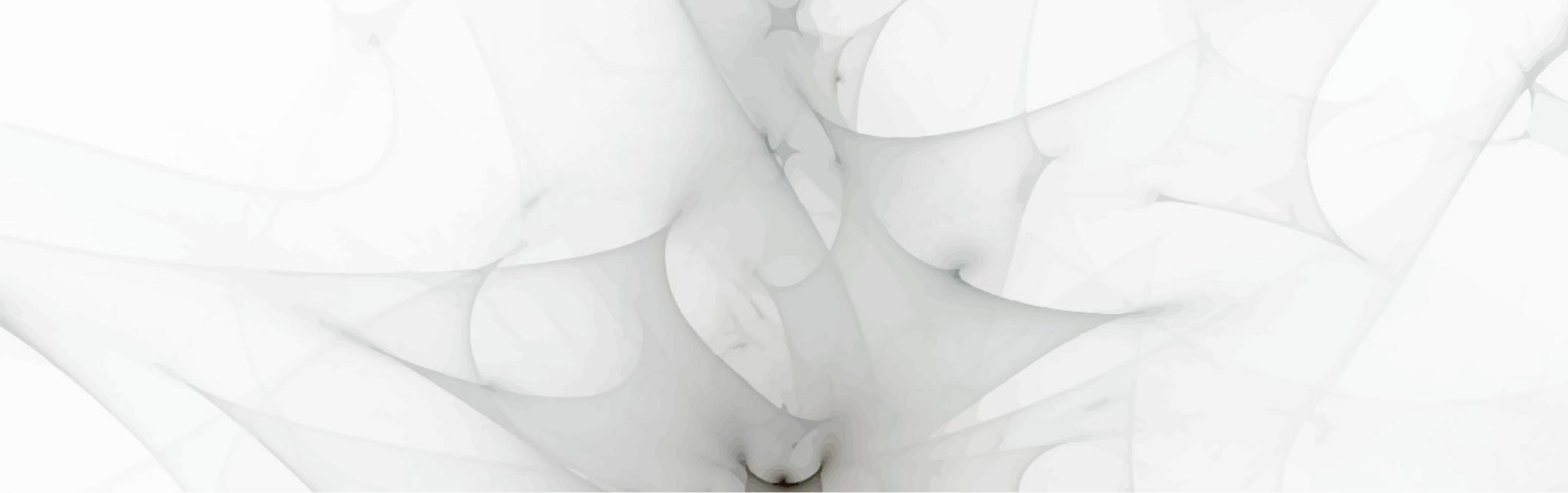
Download diagram

SVG single XHTML page with embedded SVG graphics

PNG zip file containing HTML and linked PNG Images

Download

```
StringLiteral
  ::= """ [^"]* """
     | ''' [^']* '''
    /* ws: explicit */
```



Merci de votre attention

