Snakebitten

Danger and Misfortune in the Evolution of Languages

Snakebitten:

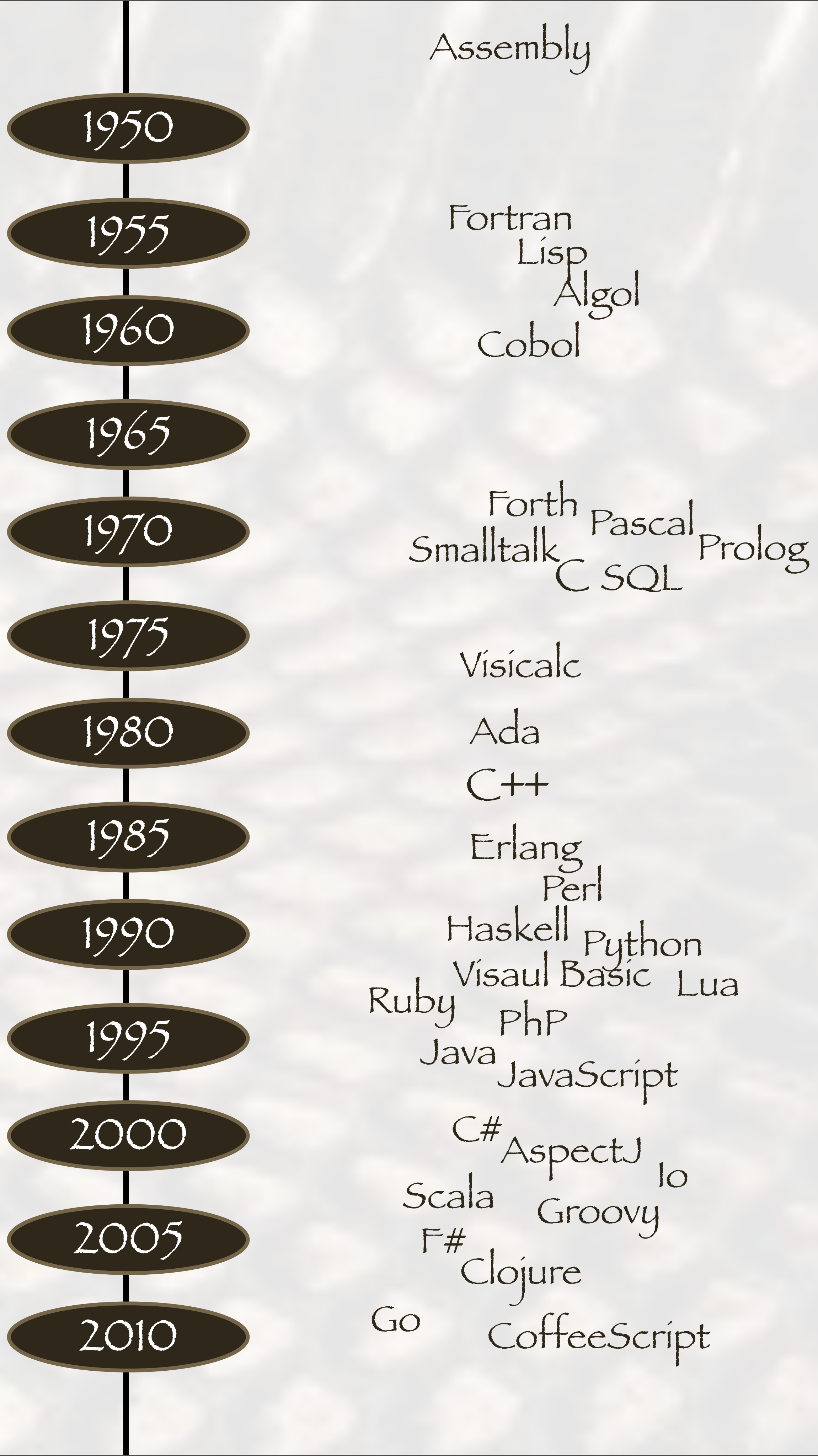
Experiencing a period of misfortune or inability to succeed.

Bruce's Extension:

due to a SOLUTION

to a PROBLEM

with UNINTENDED CONSEQUENCES



Assembly

1950

Fortran

1955

Lisp

Algol

1960

Cobol

1965

Forth

Pascal

1970

Smalltalk

Prolog

C SQL

1975

Visicalc

1980

Ada

C++

1985

Erlang

Perl

Haskell

1990

Python

Visaul Basic

Lua

Ruby

PhP

1995

Java

JavaScript

C#

2000

AspectJ

Io

Scala

Groovy

2005

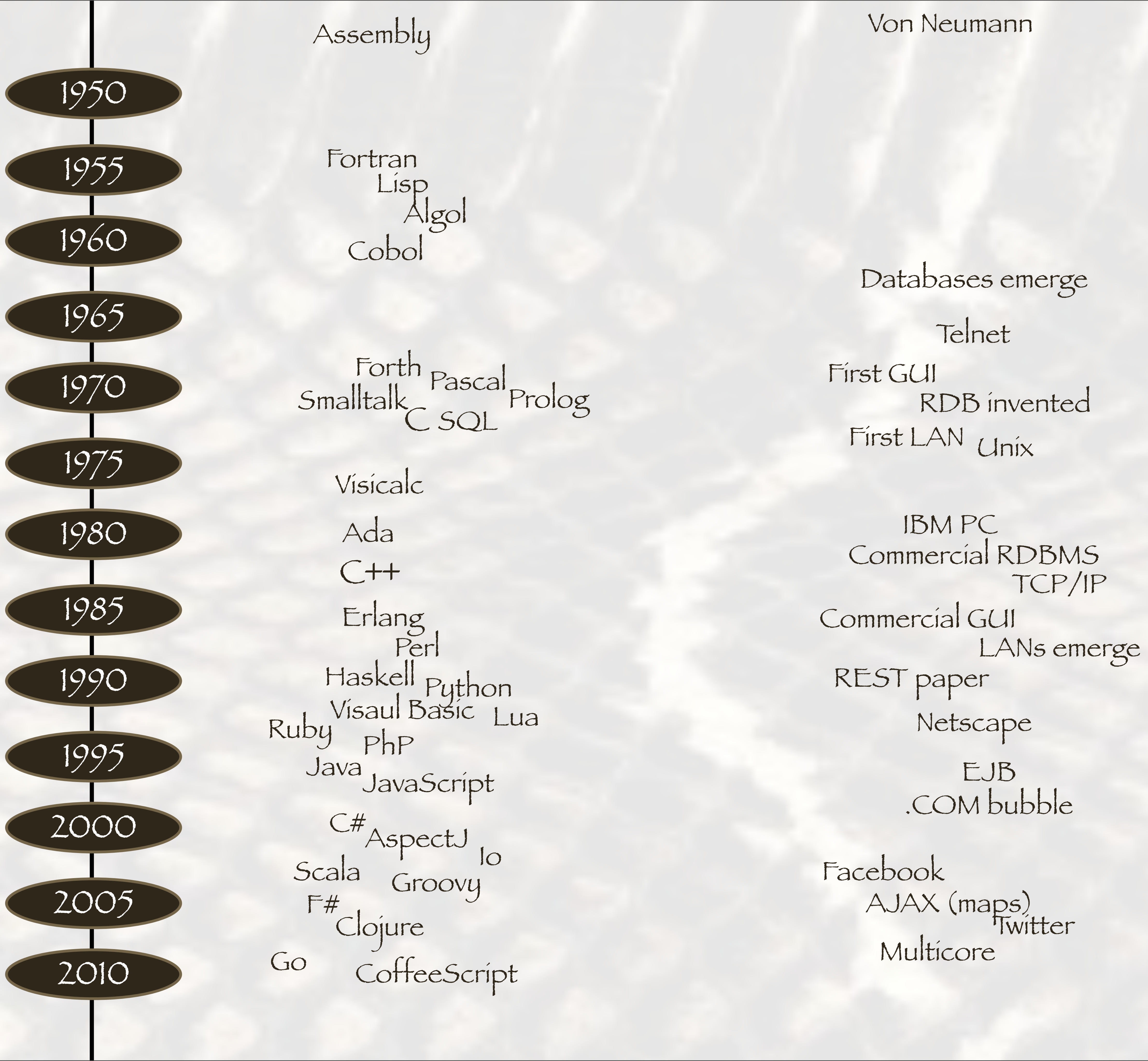
F#

Clojure

Go

2010

CoffeeScript



1950

1955

1960

1965

1970

1975

1980

1985

1990

1995

2000

2005

2010

Assembly

Fortran

Lisp

Algol

Cobol

Forth

Pascal

Smalltalk

Prolog

C

SQL

Visicalc

Ada

C++

Erlang

Perl

Haskell

Python

Visaul Basic

Lua

Ruby

PhP

Java

JavaScript

C#

AspectJ

Io

Scala

Groovy

F#

Clojure

Go

CoffeeScript

Von Neumann

Databases emerge

Telnet

First GUI

RDB invented

First LAN

Unix

IBM PC

Commercial RDBMS

TCP/IP

Commercial GUI

LANs emerge

REST paper

Netscape

EJB

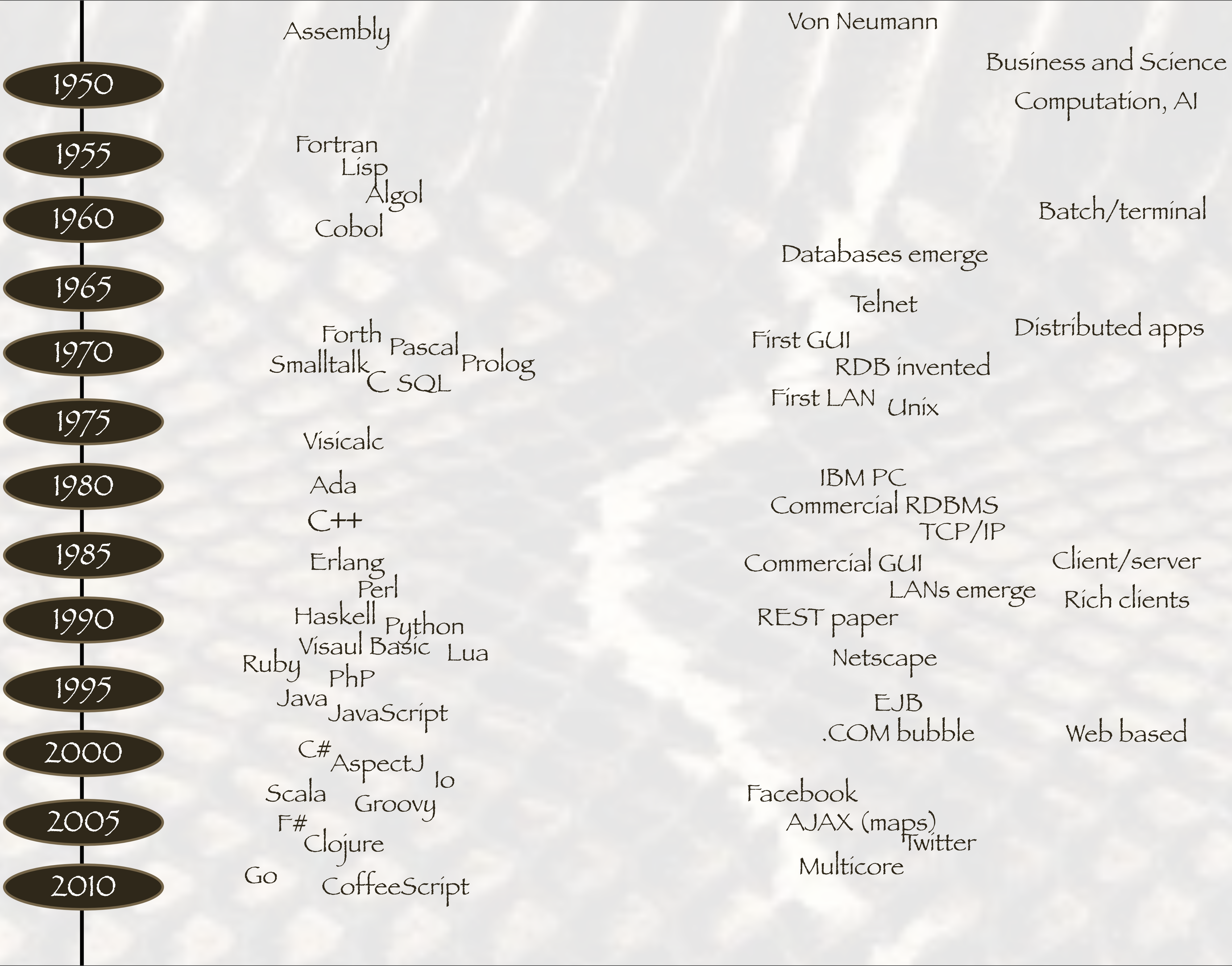
.COM bubble

Facebook

AJAX (maps)

Twitter

Multicore

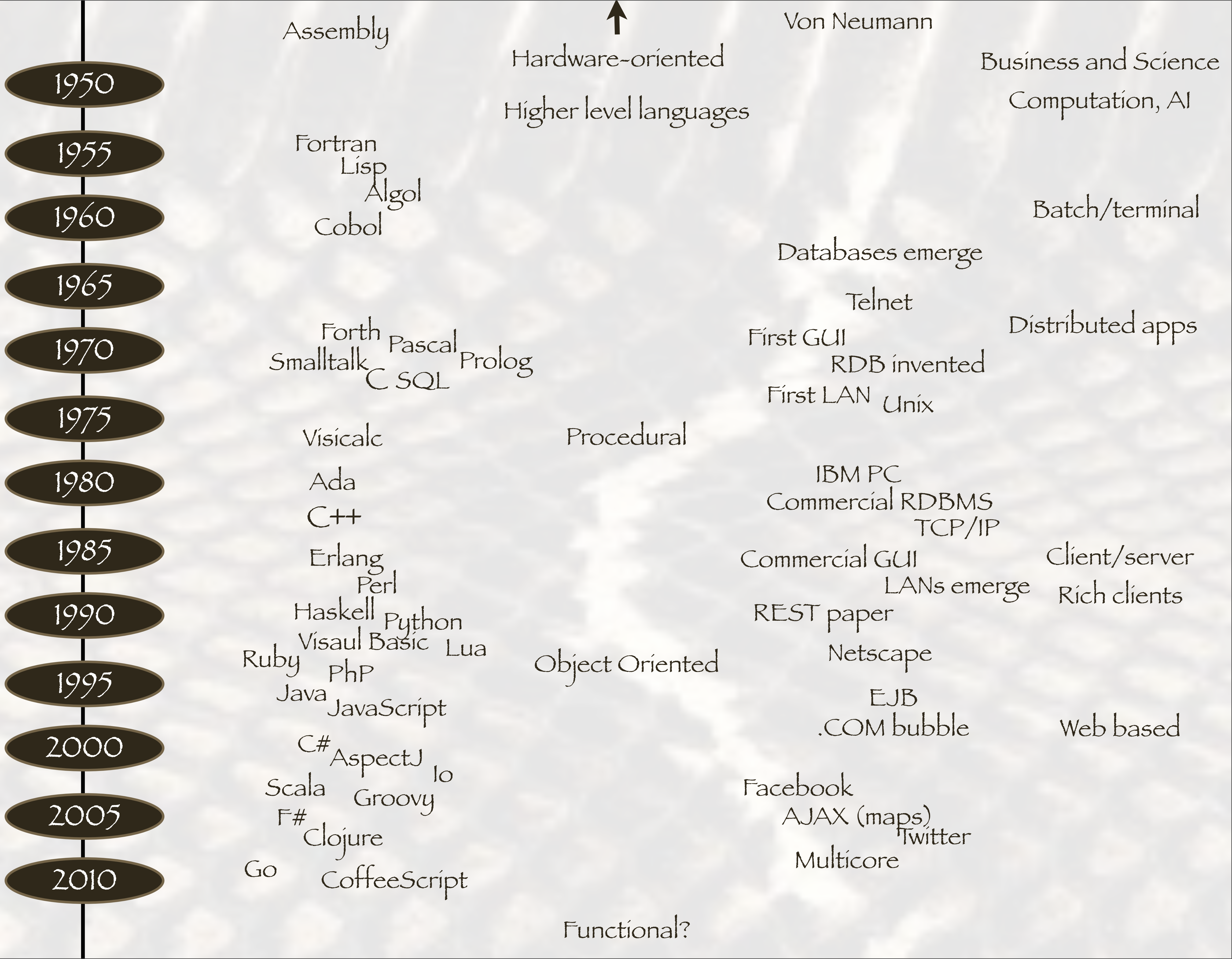


1950
1955
1960
1965
1970
1975
1980
1985
1990
1995
2000
2005
2010

Assembly
Fortran
Lisp
Algol
Cobol
Forth
Pascal
Smalltalk
C
SQL
Prolog
Visicalc
Ada
C++
Erlang
Perl
Haskell
Python
Visaul Basic
Lua
Ruby
PhP
Java
JavaScript
C#
AspectJ
Io
Scala
Groovy
F#
Clojure
Go
CoffeeScript

Von Neumann
Databases emerge
Telnet
First GUI
RDB invented
First LAN
Unix
IBM PC
Commercial RDBMS
TCP/IP
Commercial GUI
LANs emerge
REST paper
Netscape
EJB
.COM bubble
Facebook
AJAX (maps)
Twitter
Multicore

Business and Science
Computation, AI
Batch/terminal
Distributed apps
Client/server
Rich clients
Web based



Assembly

Hardware-oriented

Von Neumann

Business and Science

1950

Higher level languages

Computation, AI

Fortran

1955

Lisp

Algol

1960

Batch/terminal

Cobol

Databases emerge

1965

Telnet

Forth

Pascal

First GUI

Distributed apps

1970

Smalltalk

Prolog

RDB invented

C

SQL

First LAN

Unix

1975

Visicalc

Procedural

IBM PC

1980

Ada

Commercial RDBMS

C++

TCP/IP

Erlang

1985

Commercial GUI

Client/server

Perl

LANs emerge

Rich clients

Haskell

REST paper

1990

Python

Visaul Basic

Lua

Ruby

Netscape

Object Oriented

PhP

1995

Java

EJB

JavaScript

.COM bubble

Web based

C#

2000

AspectJ

Io

Scala

Facebook

Groovy

2005

F#

AJAX (maps)

Twitter

Clojure

Multicore

Go

2010

CoffeeScript

Functional?

1950

1955

1960

1965

1970

1975

1980

1985

1990

1995

2000

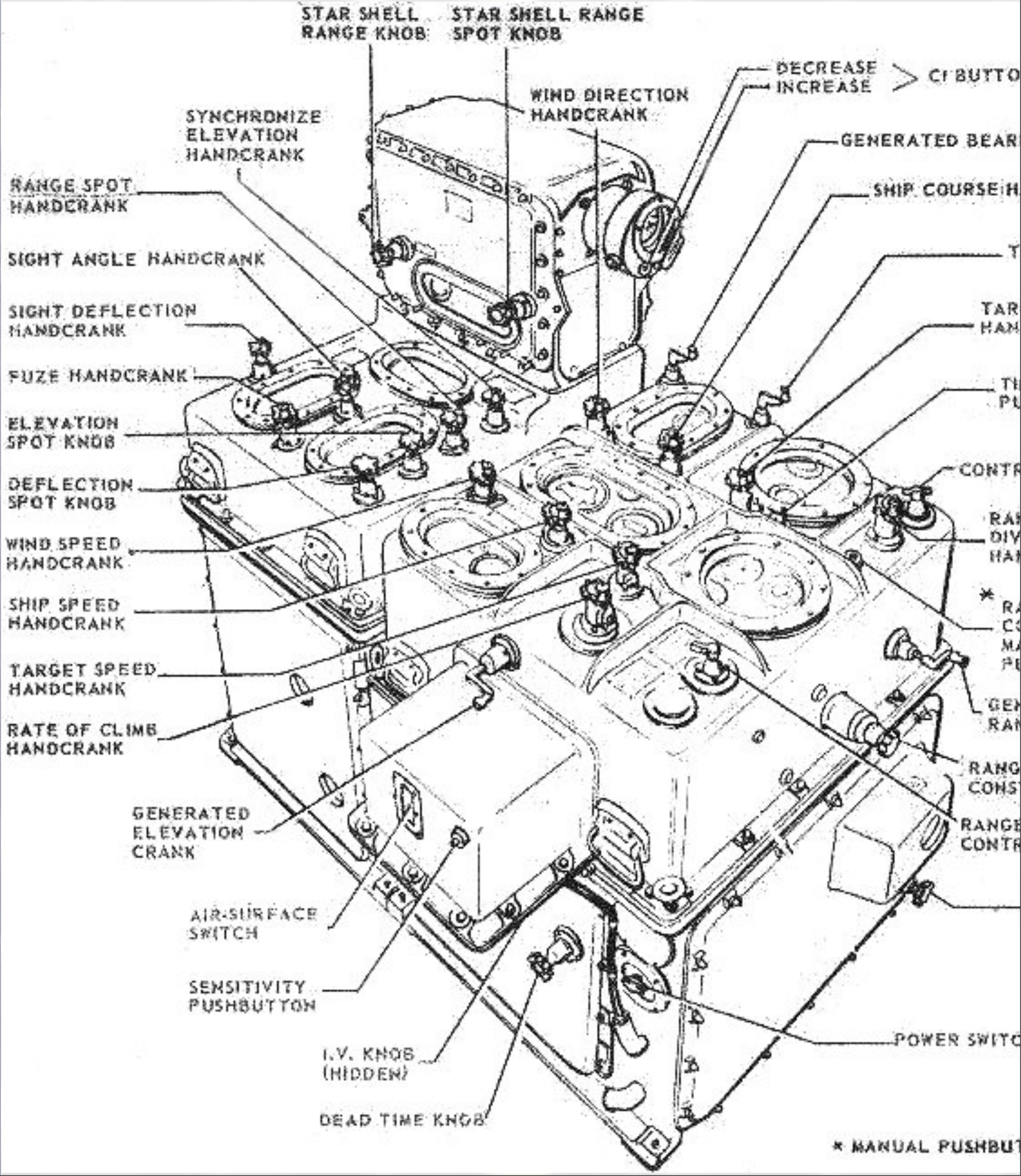
2005

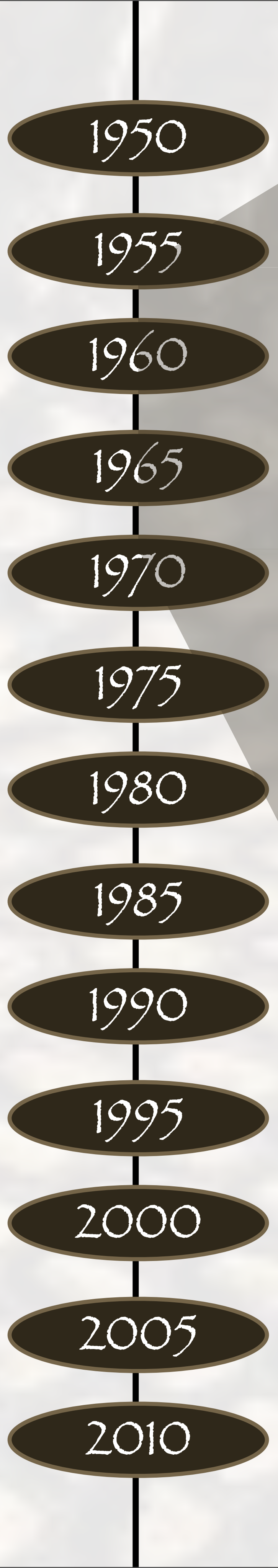
2010

Assembly

Hardware-oriented languages

Mark I Computer





1950
1955
1960
1965
1970
1975
1980
1985
1990
1995
2000
2005
2010

HIGH LEVEL LANGUAGES

Fortran

Algol

Lisp

Cobol

2 Tier ARCHITECTURE

Business,

Science,

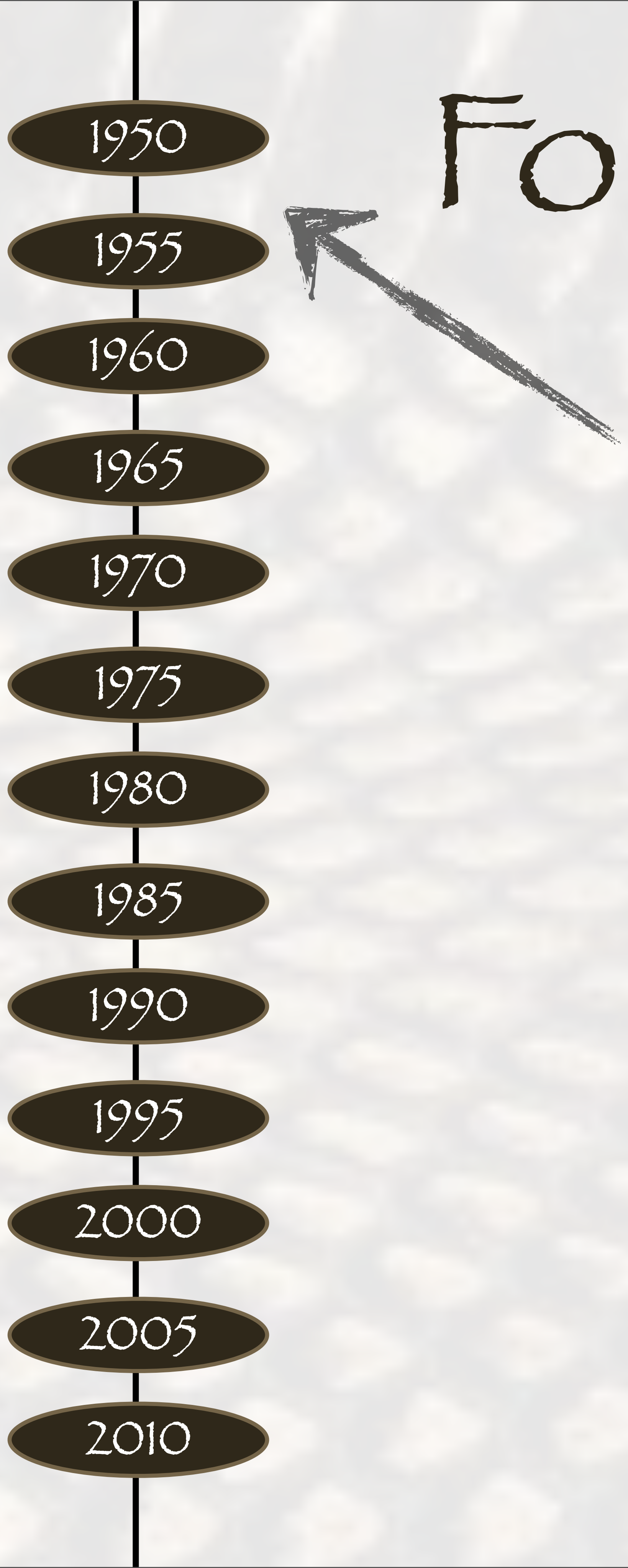
AI

Databases

Fortran, COBOL

VS

LISP



1950

1955

1960

1965

1970

1975

1980

1985

1990

1995

2000

2005

2010

DEATHMATCH

LISP

FORTRAN, COBOL

(LISP)

FORTRAN, COBOL

(　　LISP　　)

FORTRAN, COBOL

(LISP)

FORTRAN, COBOL

(LISP)

FORTRAN, COBOL

(LISP)

FORTRAN, COBOL

(lisp)

FORTRAN, COBOL

(||||)

FORTRAN, COBOL

0

FORTRAN, COBOL

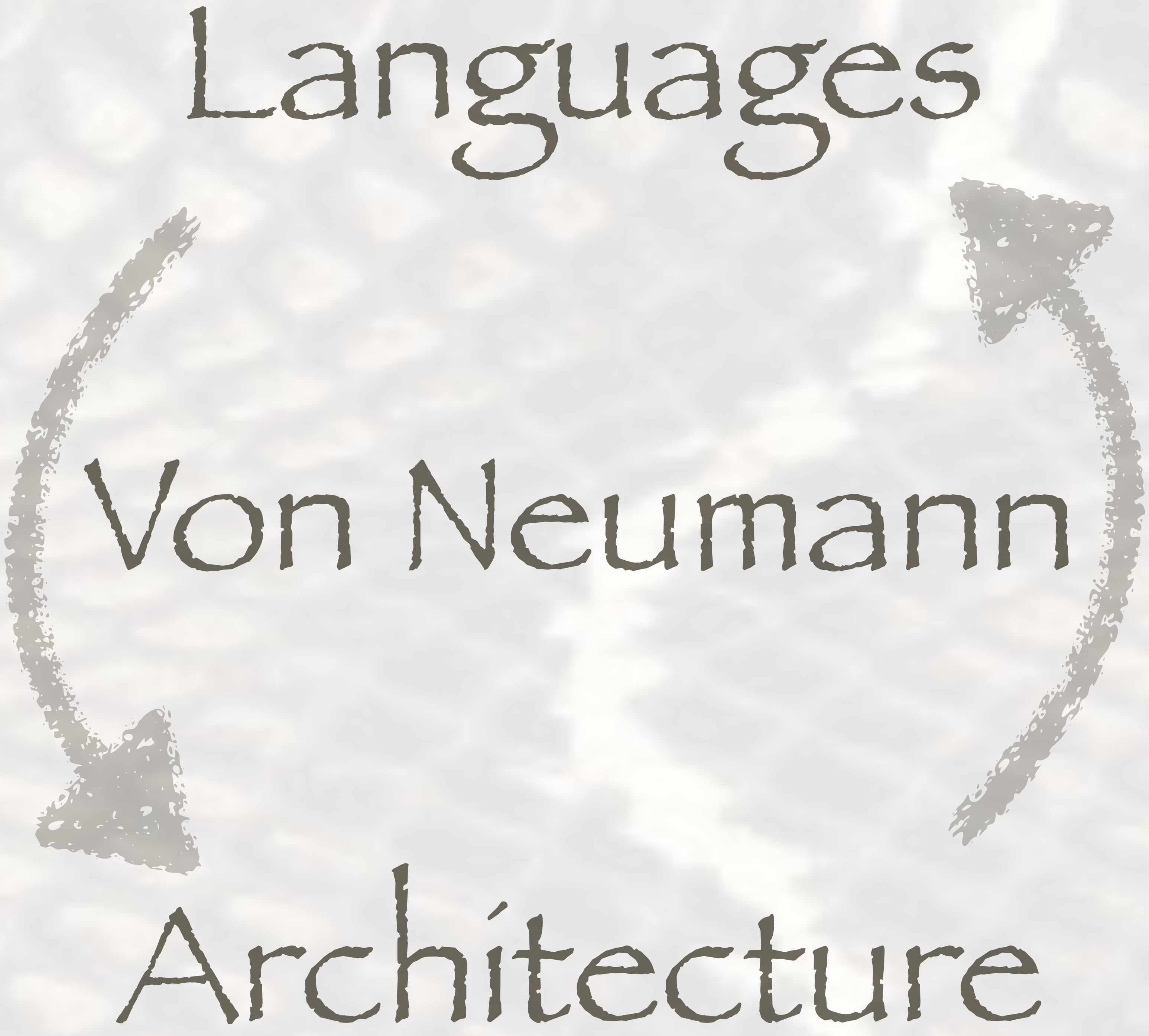
FORTRAN, COBOL

```
    C     SWAP X,Y
    C
100       TEMP = X
          X = Y
          Y = TEMP
```

changing values

mutable state

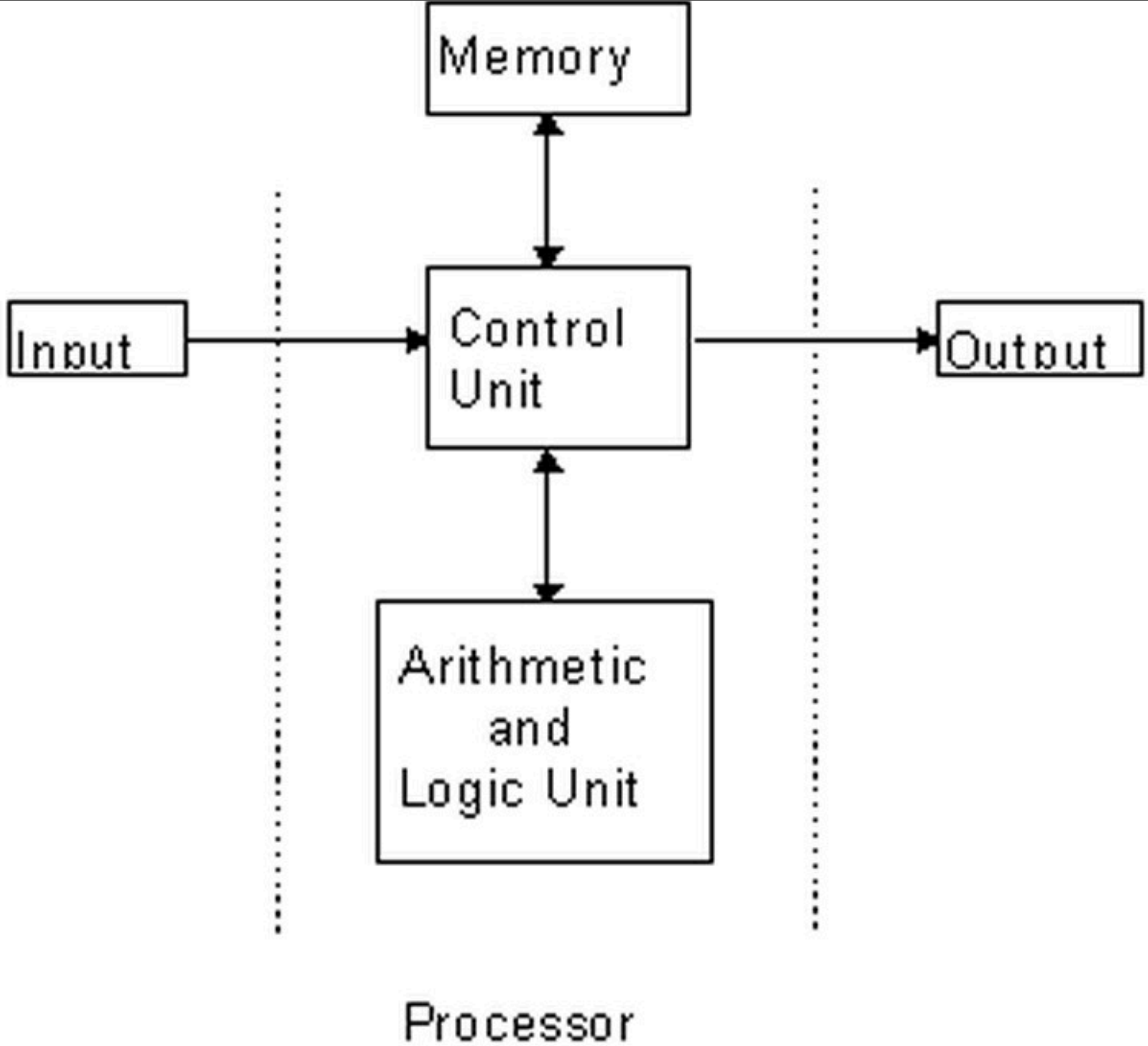
binding Identity to Value



Languages

Von Neumann

Architecture



Memory

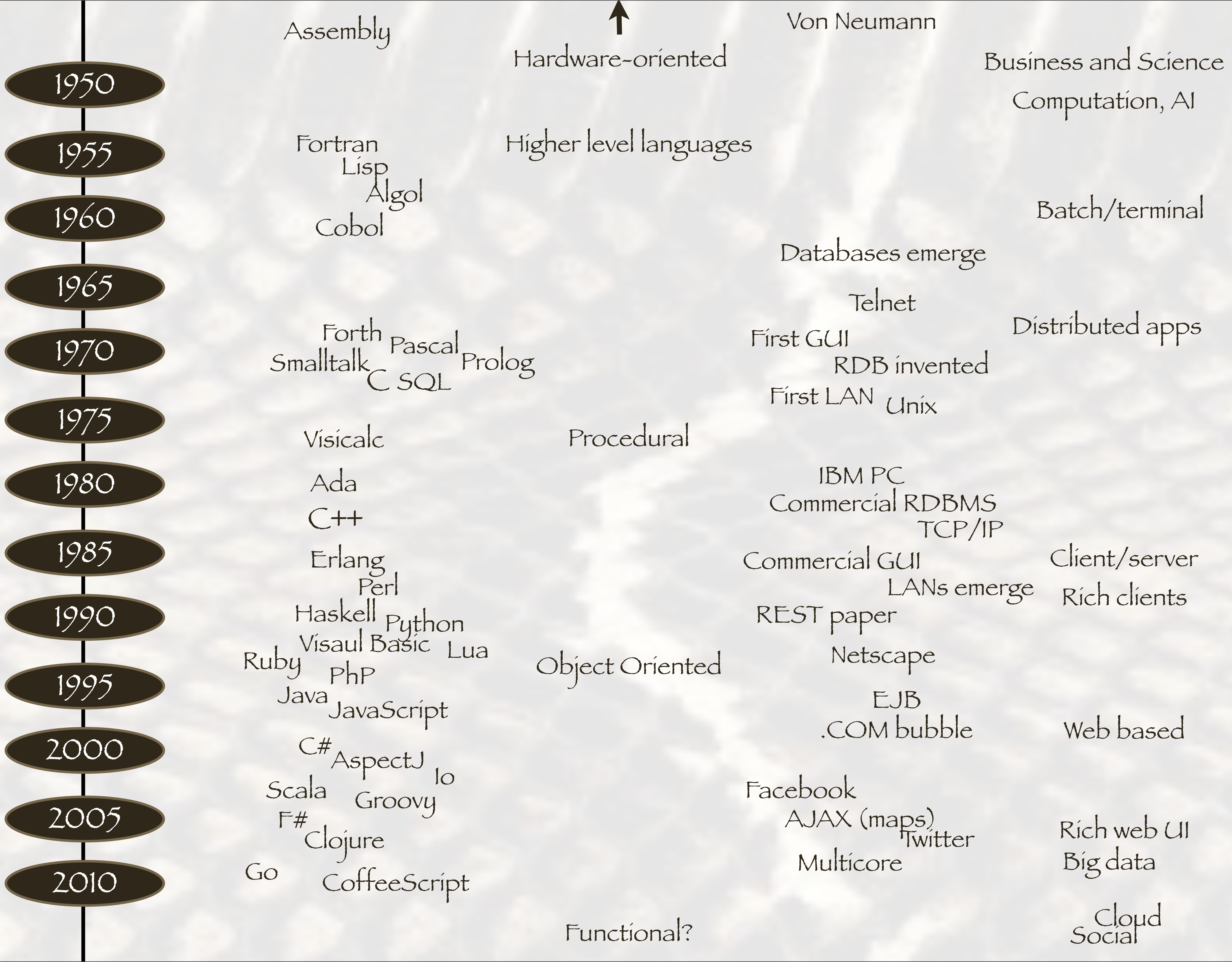
Input

Control Unit

Output

Arithmetic and Logic Unit

Processor



Assembly
Hardware-oriented
Von Neumann
Business and Science
Computation, AI

1950

1955

Fortran
Lisp
Algol
Higher level languages

1960

Cobol
Batch/terminal
Databases emerge

1965

Telnet

1970

Forth
Pascal
Smalltalk
C
SQL
Prolog
First GUI
RDB invented
First LAN
Unix
Distributed apps

1975

Visicalc
Procedural

1980

Ada
C++
IBM PC
Commercial RDBMS
TCP/IP

1985

Erlang
Perl
Commercial GUI
LANs emerge
Client/server
Rich clients

1990

Haskell
Python
Visaul Basic
Lua
Ruby
PhP
REST paper
Object Oriented
Netscape

1995

Java
JavaScript
EJB
.COM bubble
Web based

2000

C#
AspectJ
Io
Scala
Groovy
Facebook

2005

F#
Clojure
AJAX (maps)
Twitter
Multicore
Rich web UI
Big data

2010

Go
CoffeeScript
Functional?
Cloud
Social

1950

1955

1960

1965

1970

1975

1980

1985

1990

1995

2000

2005

2010

Procedural

Forth

Pascal

Smalltalk

Prolog

C

SQL

Visicalc

Ada

C++

CLIENT-SERVER

Telnet

First GUI

RDB invented

First LAN

Unix

IBM PC

Commercial RDBMS

DEATHMATCH

```
Transcript show: 'hello world'.
```

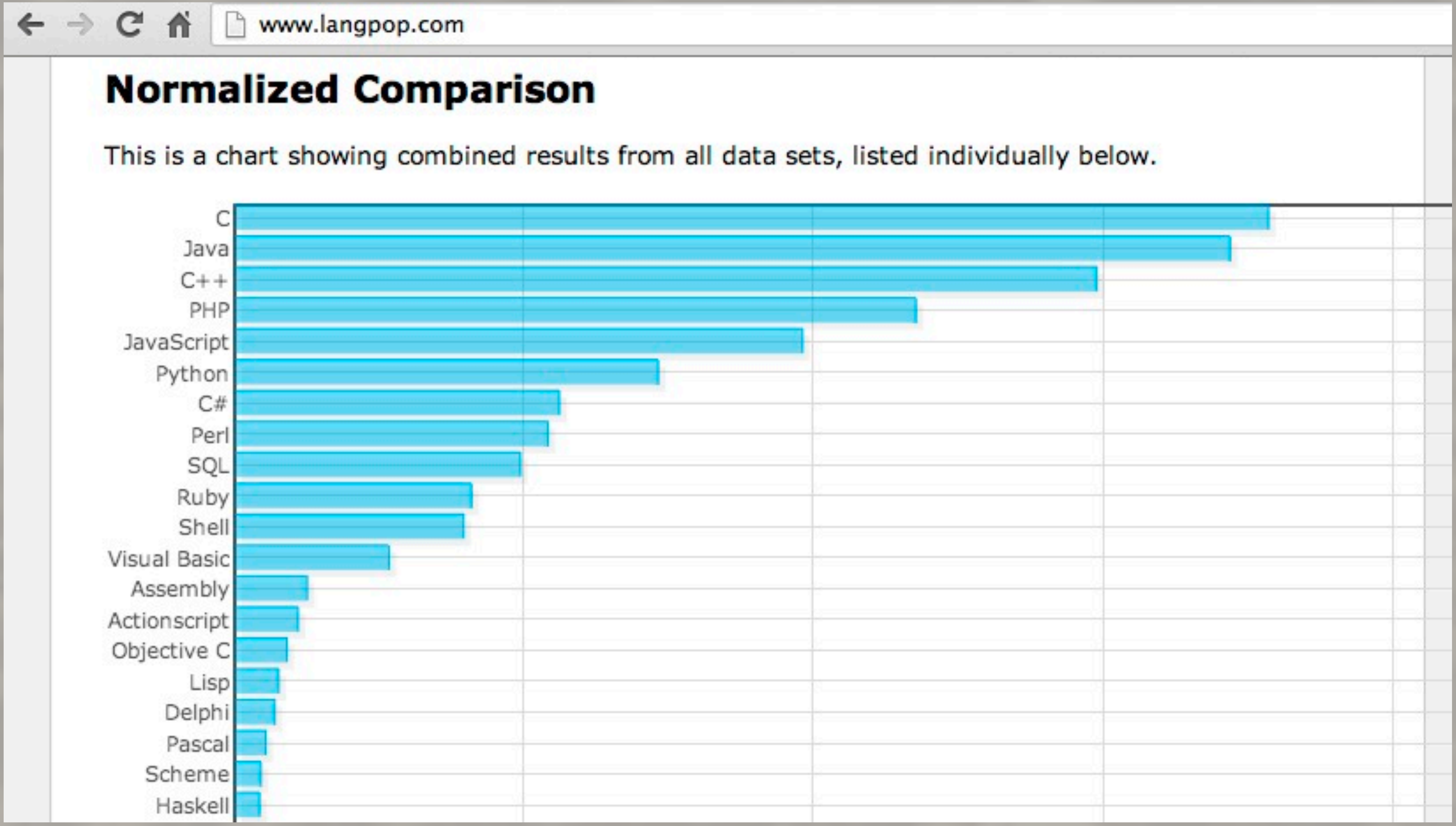
```
#include<stdio.h>
main()
{
    printf("Hello World");
}
```

C Smalltalk

Smalltalk

C

langpop.com



Normalized Comparison

This is a chart showing combined results from all data sets, listed individually below.

- Explicit allocation
- Bit-level control
- Weak, flexible types

+ Garbage collection
+ String manipulation
+ Reliability

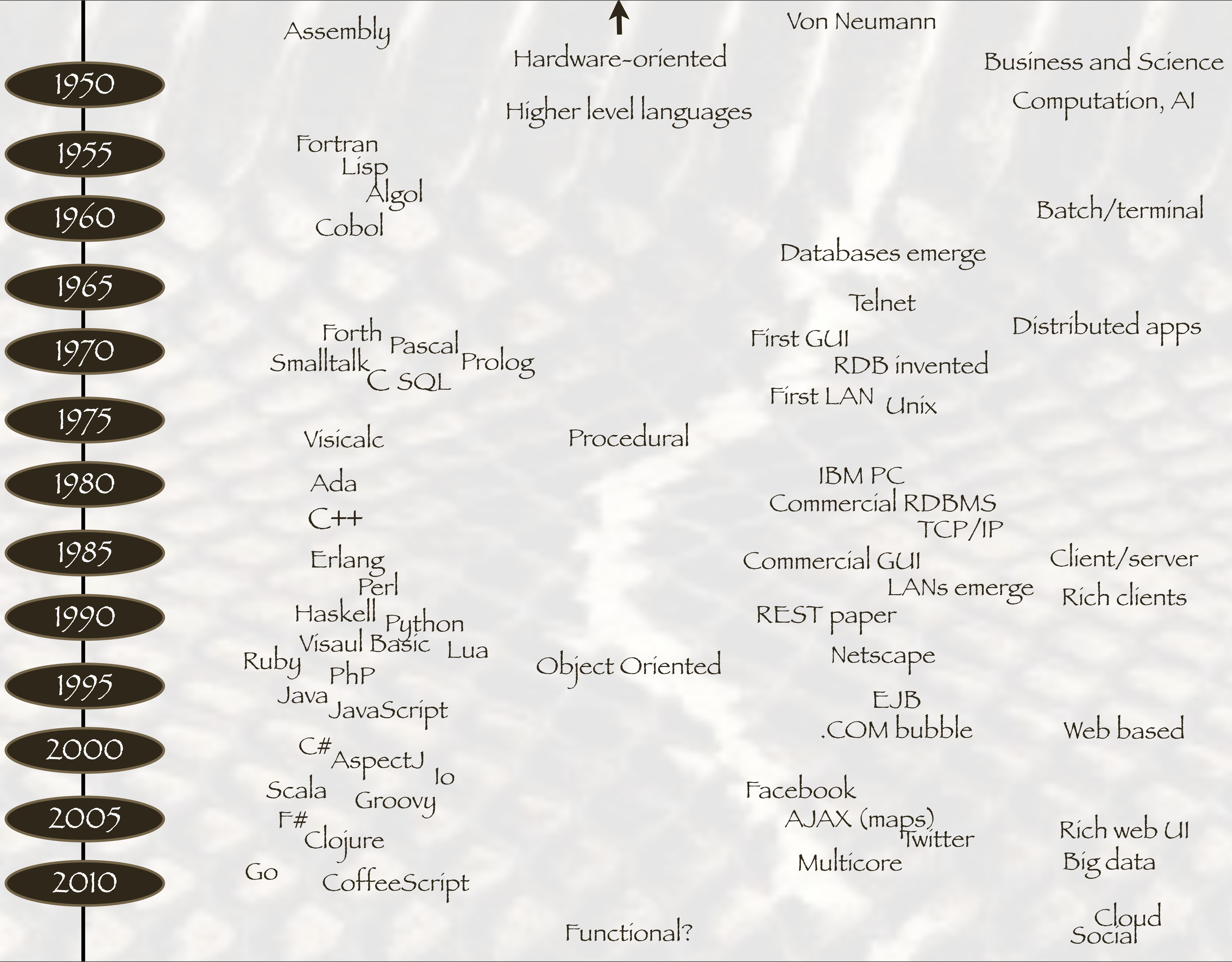
– Control

+Productivity

– Performance

+Expressiveness

- Paradigm (imperative)
- Syntax
- Type system
- Idioms



Assembly Hardware-oriented Von Neumann

1950 Business and Science

Higher level languages Computation, AI

1955 Fortran

Lisp

Algol

1960 Batch/terminal

Cobol

Databases emerge

1965

Telnet

Forth Pascal First GUI Distributed apps

1970 Smalltalk C SQL Prolog RDB invented

First LAN Unix

1975 Visicalc Procedural

IBM PC

1980 Ada Commercial RDBMS

C++ TCP/IP

1985 Erlang Commercial GUI Client/server

Perl LANs emerge Rich clients

1990 Haskell Python REST paper

Visaul Basic Lua

Ruby Object Oriented Netscape

1995 PhP

Java JavaScript EJB

.COM bubble Web based

2000 C# AspectJ

Io

Scala Groovy Facebook

2005 F# AJAX (maps) Twitter Rich web UI

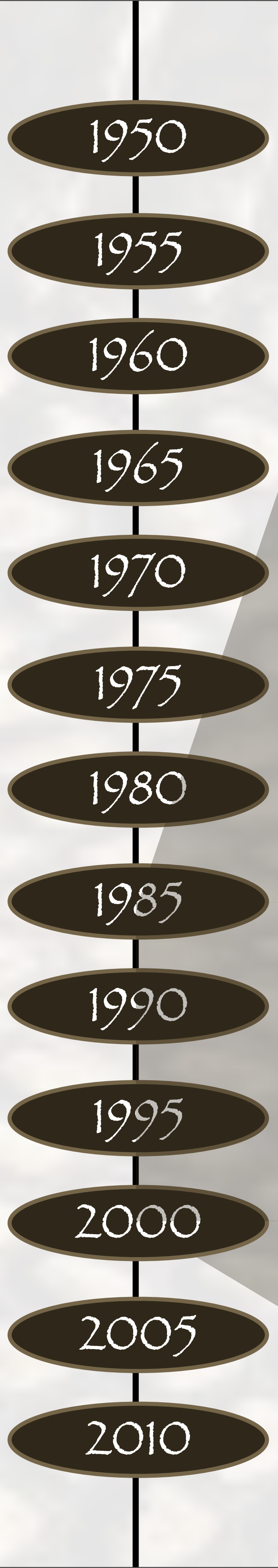
Clojure

Go Multicore Big data

2010 CoffeeScript

Functional? Cloud

Social



1950
1955
1960
1965
1970
1975
1980
1985
1990
1995
2000
2005
2010

Object Oriented

Erlang
Perl
Haskell
Python
Visaul Basic
Lua
Ruby
PhP
Java
JavaScript
C#

Web-based

REST paper

Netscape

EJB

.COM bubble

TCP/IP

Deployment

- Deployment
- Complexity
- Costs

DEATHMATCH

Java

Smalltalk

Smalltalk

Java

Smalltalk

||

Java

X

Java

Java Visual Basic

Visual Basic

Java

Visual Basic

||

Java

X

Java

Java

Java TRTFTJ

TRTFTJ

Java

TRTFTJ

||

Java

X

Java

Java

Java

- Static
- Explicit
- Nominal

- Static (dynamic)
- Explicit (inferred)
- Nominal (structural)

- When do you bind?
- How much do you type?
- What do you mean?

- Component models
- Containers
- Specialty

```
class Nil
  def blank?
    false
  end
end
```

```
class String
  def blank?
    self == ""
  end
end
```

```
class Object
  def blank?
    true
  end
end
```

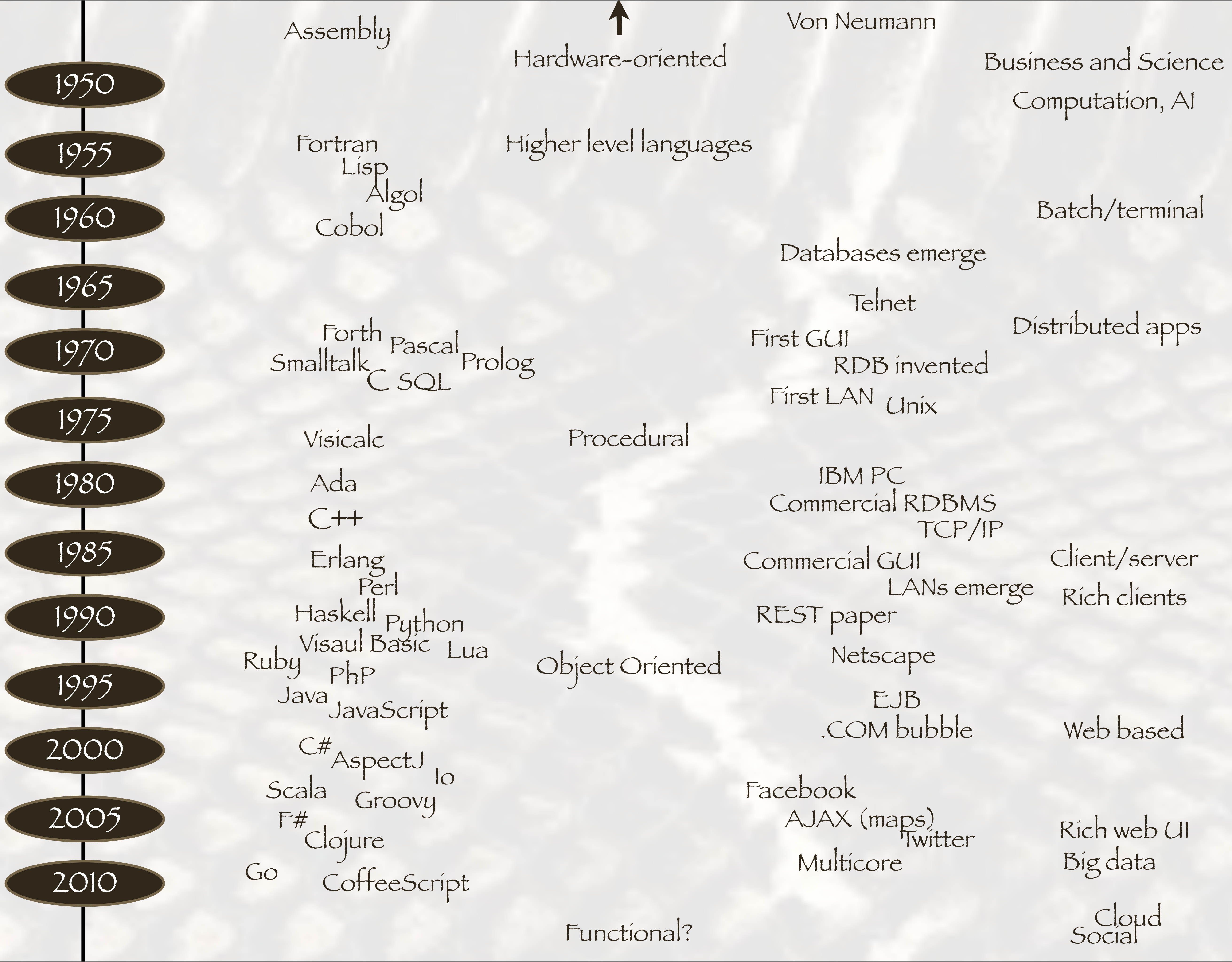
```
"".blank?
```

Java

Java

Java

Java



Assembly
↑
Von Neumann

Hardware-oriented

Business and Science

1950

Computation, AI

Fortran

Higher level languages

1955

Lisp

Algol

Batch/terminal

1960

Cobol

Databases emerge

1965

Telnet

Forth

Distributed apps

First GUI

1970

Pascal

Smalltalk

Prolog

RDB invented

C SQL

First LAN

Unix

1975

Visicalc

Procedural

IBM PC

1980

Ada

Commercial RDBMS

C++

TCP/IP

1985

Erlang

Commercial GUI

Client/server

Perl

LANs emerge

Rich clients

Haskell

REST paper

1990

Python

Visaul Basic

Lua

Ruby

Object Oriented

Netscape

PhP

1995

Java

JavaScript

EJB

2000

.COM bubble

Web based

C#

AspectJ

Io

Scala

Facebook

Groovy

F#

AJAX (maps)

Rich web UI

2005

Twitter

Clojure

Multicore

Big data

Go

2010

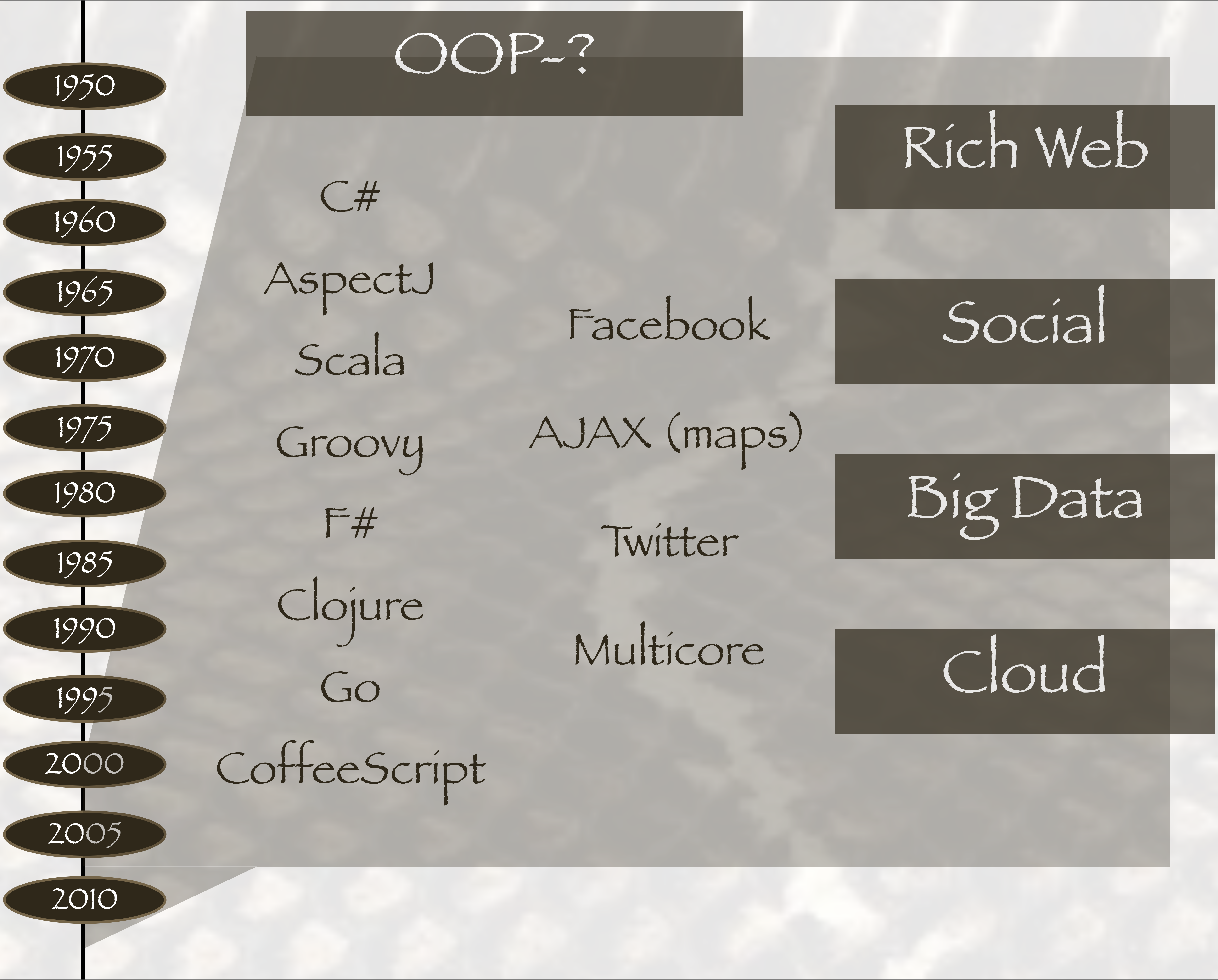
CoffeeScript

Cloud

Functional?

Social

OOP-?



1950

1955

1960

1965

1970

1975

1980

1985

1990

1995

2000

2005

2010

C#

AspectJ

Scala

Groovy

F#

Clojure

Go

CoffeeScript

Facebook

AJAX (maps)

Twitter

Multicore

Rich Web

Social

Big Data

Cloud

Multicore

Functional

Integration

Why do languages emerge?

They solve A PROBLEM.

Why snakebites?

We apply a SPECIFIC solution to a GENERAL problem

What can we do?

Sell

Embrace

Love

?