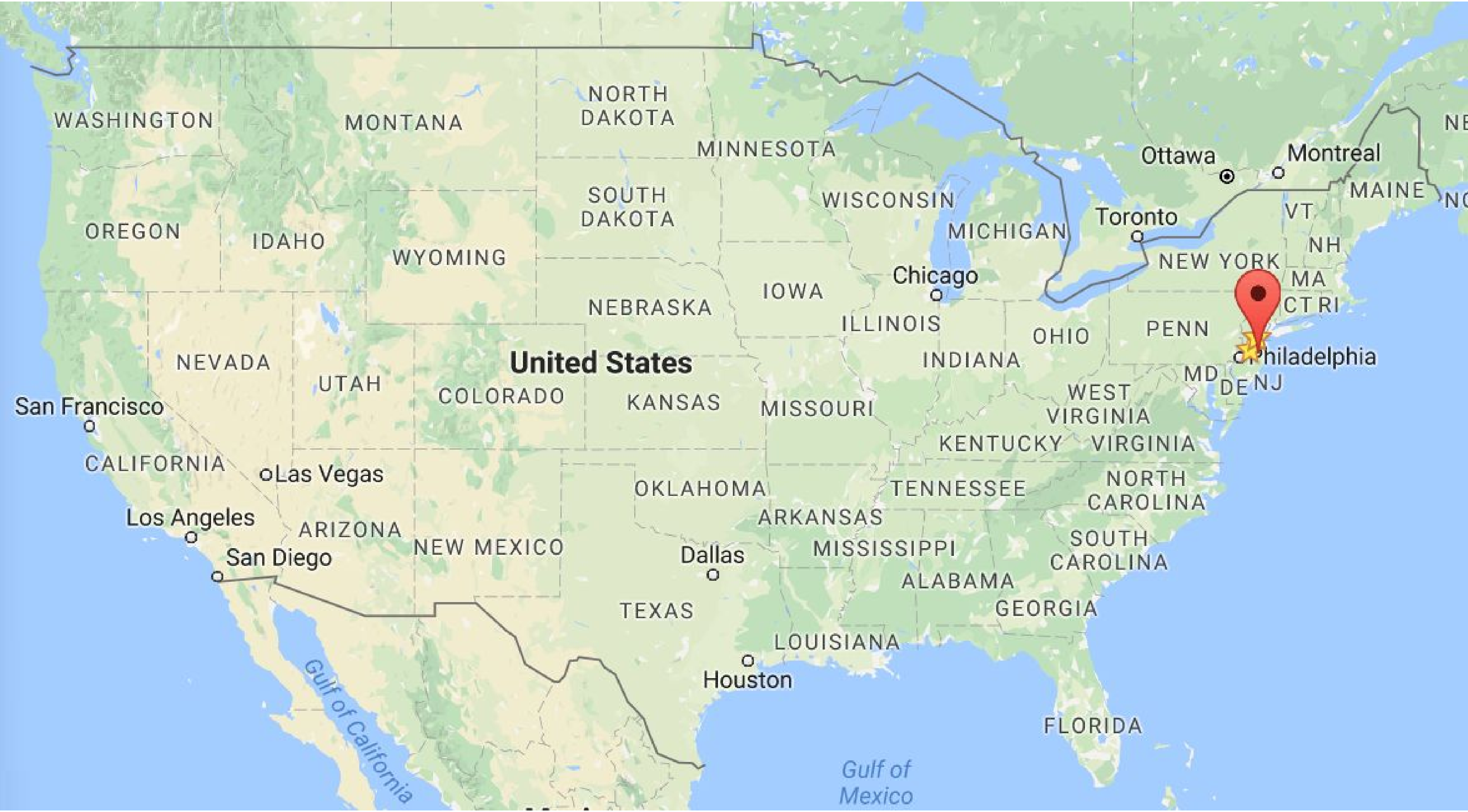
Hobbyist APLing in the 21st Century.

Alex Weiner (New Jersey, USA)


NORTH DAKOTA
WASHINGTON
MONTANA
MINNESOTA
Ottawa
Montreal
MAINE
SOUTH DAKOTA
WISCONSIN
VT
OREGON
IDAHO
MICHIGAN
Toronto
NH
WYOMING
NEW YORK
Chicago
MA
IOWA
CT RI
NEBRASKA
ILLINOIS
PENN
OHIO
NEVADA
United States
INDIANA
Philadelphia
UTAH
MD
COLORADO
KANSAS
DE NJ
San Francisco
MISSOURI
WEST VIRGINIA
KENTUCKY
VIRGINIA
CALIFORNIA
Las Vegas
OKLAHOMA
TENNESSEE
NORTH CAROLINA
Los Angeles
ARIZONA
ARKANSAS
NEW MEXICO
SOUTH CAROLINA
Dallas
MISSISSIPPI
San Diego
ALABAMA
GEORGIA
TEXAS
LOUISIANA
Houston
Gulf of California
FLORIDA
Gulf of Mexico

Alex Weiner

Alex Weiner

- Electrical & Computer Engineer

Alex Weiner

- Electrical & Computer Engineer
- APLer

Why APL?

- Lots of things can be modeled as a matrix

Why APL?

- Lots of things can be modeled as a matrix
  - Circuits
  - Images
  - Differential equations

Why APL?

- Lots of things can be modeled as a matrix

Why APL?

- Lots of things can be modeled as a matrix
  - Circuits
  - Images
  - Differential equations

Why APL?

- Lots of things can be modeled as a matrix
  - Circuits
  - Images
  - Differential equations
- Computer Architecture

Why APL?

- Lots of things can be modeled as a matrix
  - Circuits
  - Images
  - Differential equations
- Computer Architecture
  - “A Programming Language” By Iverson

“Hobbyist APLing in the 21st Century”

What does that even mean?

Hobbyist

- Must be fun

Hobbyist

- Must be fun
- Must be low cost

Hobbyist

- Must be fun
- Must be low cost
- Can be non-practical

Hobbyist

- Must be fun
- Must be low cost
- Can be non-practical
- Must not be frustrating

Hobbyist

- Must be fun
- Must be low cost
- Can be non-practical
- Must not be frustrating
- A learning experience

APLing

- Programming in any APL dialect or related language

APLing

- Programming in any APL dialect or related language
- Talking about any APL dialect or related language

21st Century

- A web application

21st Century

- A web application
  - Your program ultimately outputs HTML

Application development in APL

Pick an operating system

Pick an operating system

- Linux
- macOS
- Windows

Pick an APL

Pick an APL

- APL
    - Dyalog APL
    - GNU APL
    - J

Pick an APL

- APL
  - Dyalog APL
  - GNU APL
  - J

- Other
  - J
  - A+,Q,K
  - APL2000, NARS2000
  - ELI
  - S (R is S!)
  - MATLAB, Mathematica
  - Go, C++ (iota)

Determine an application

Determine an application

- A hard problem in computer programming

Determine an application

- A hard problem in computer programming
- Do something cool

Determine an application

- A hard problem in computer programming
- Do something cool
- Something that shows my friends how cool APL is

My Application

My Application

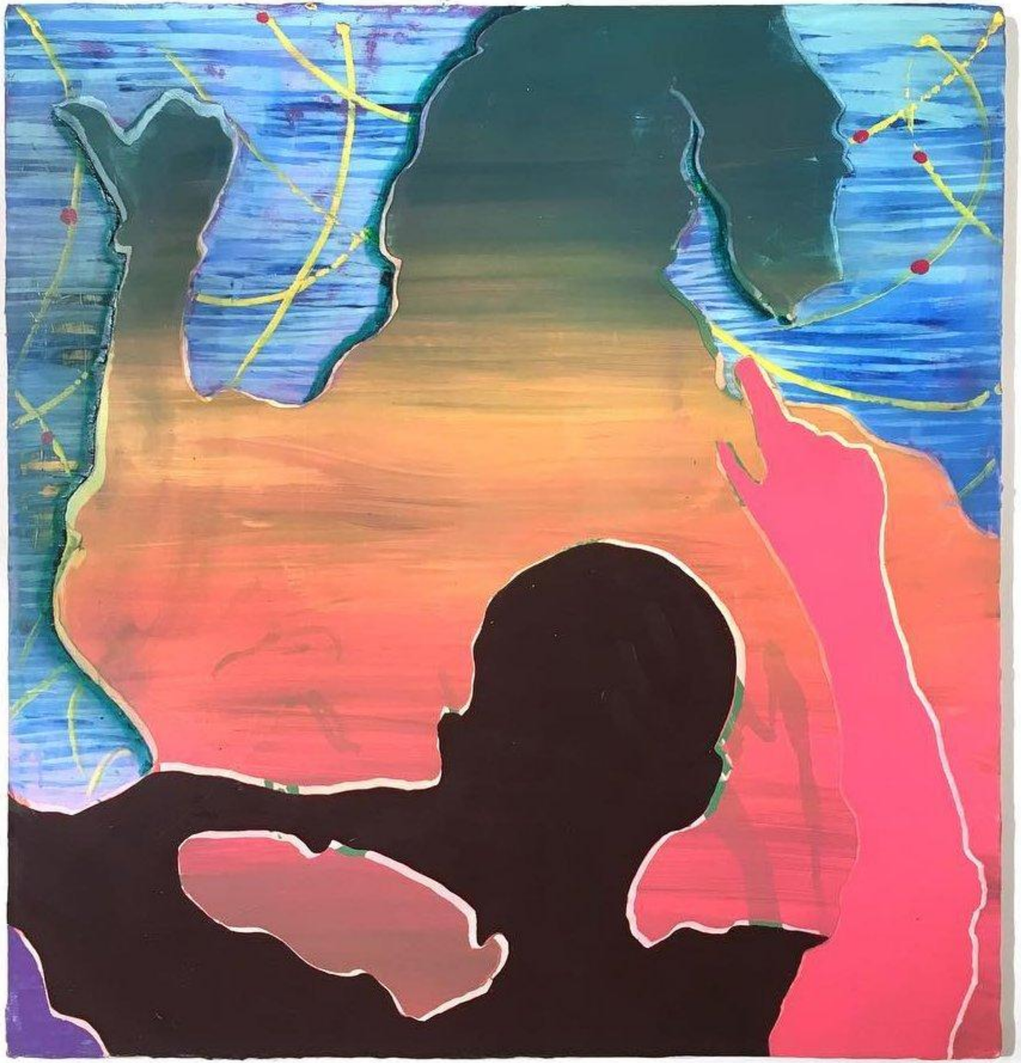
- Called “Flake”

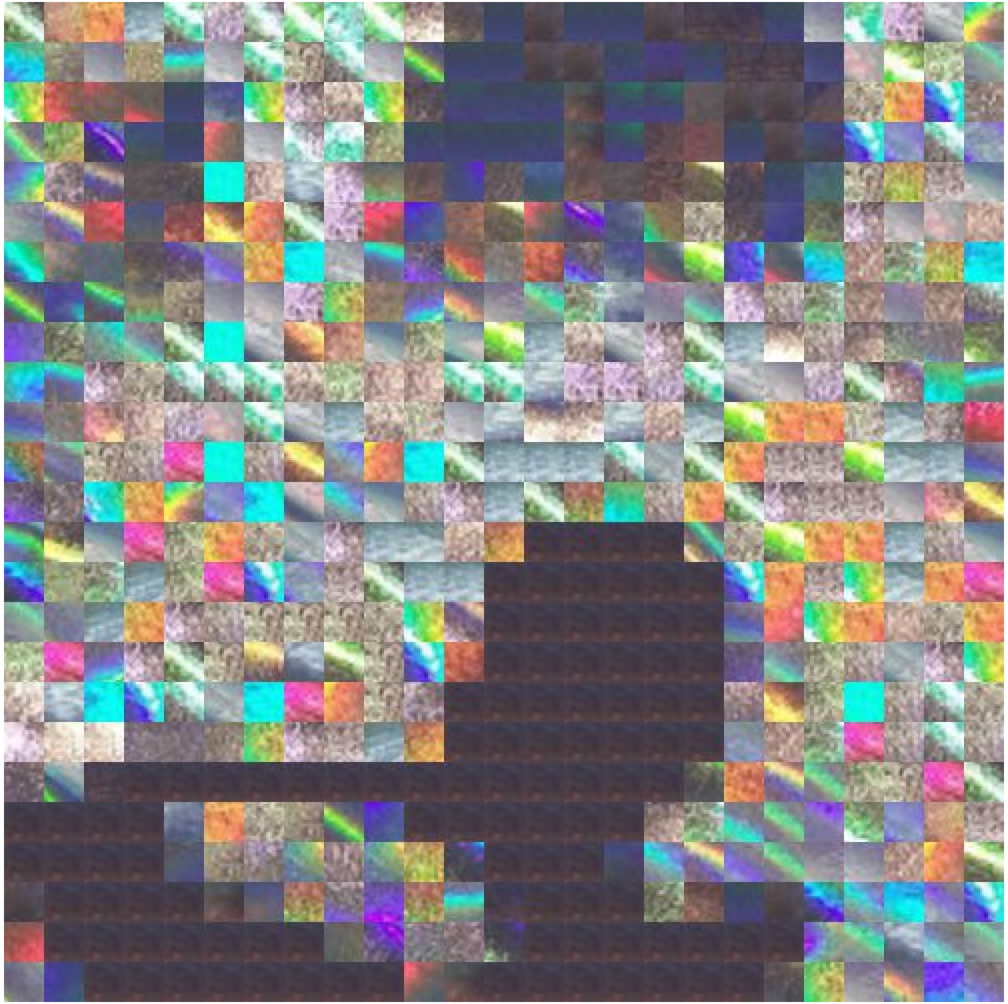
My Application

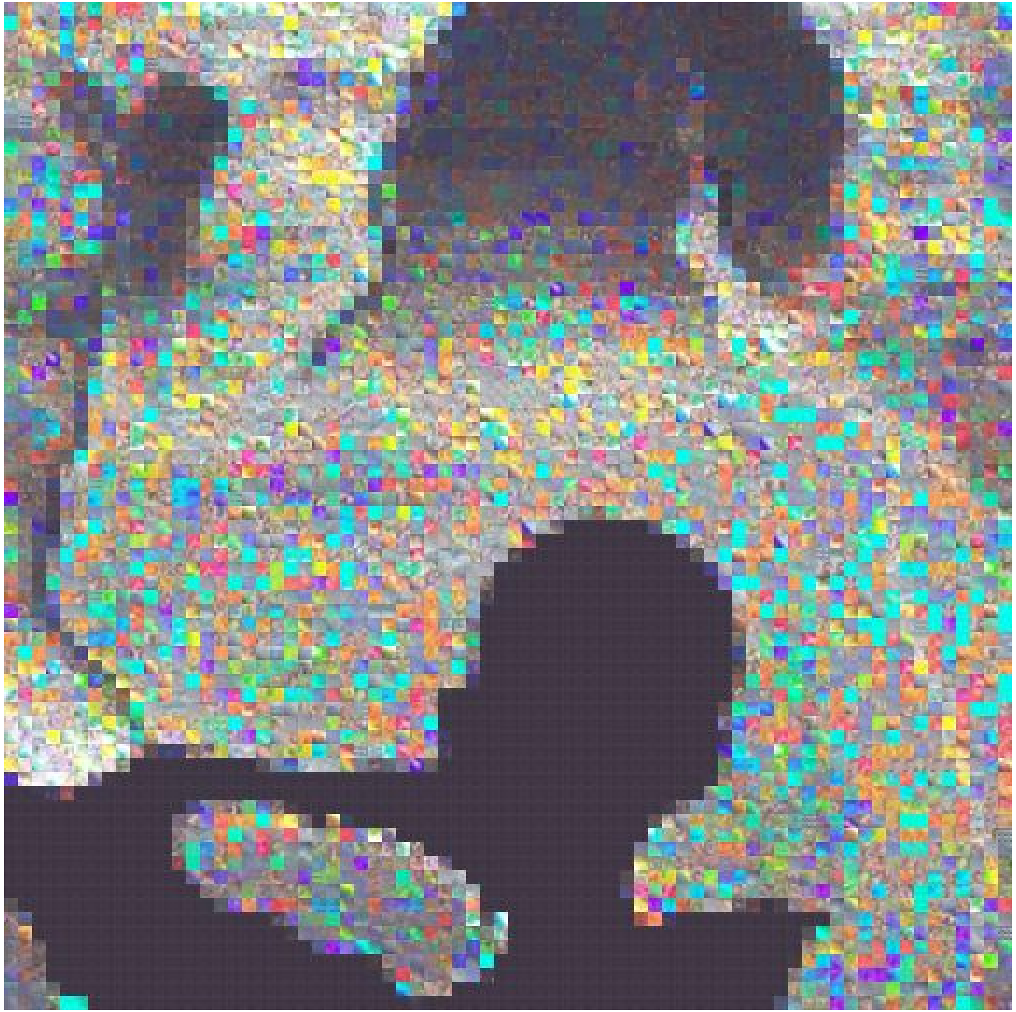
- Called “Flake”
  - Chops
  - Matches

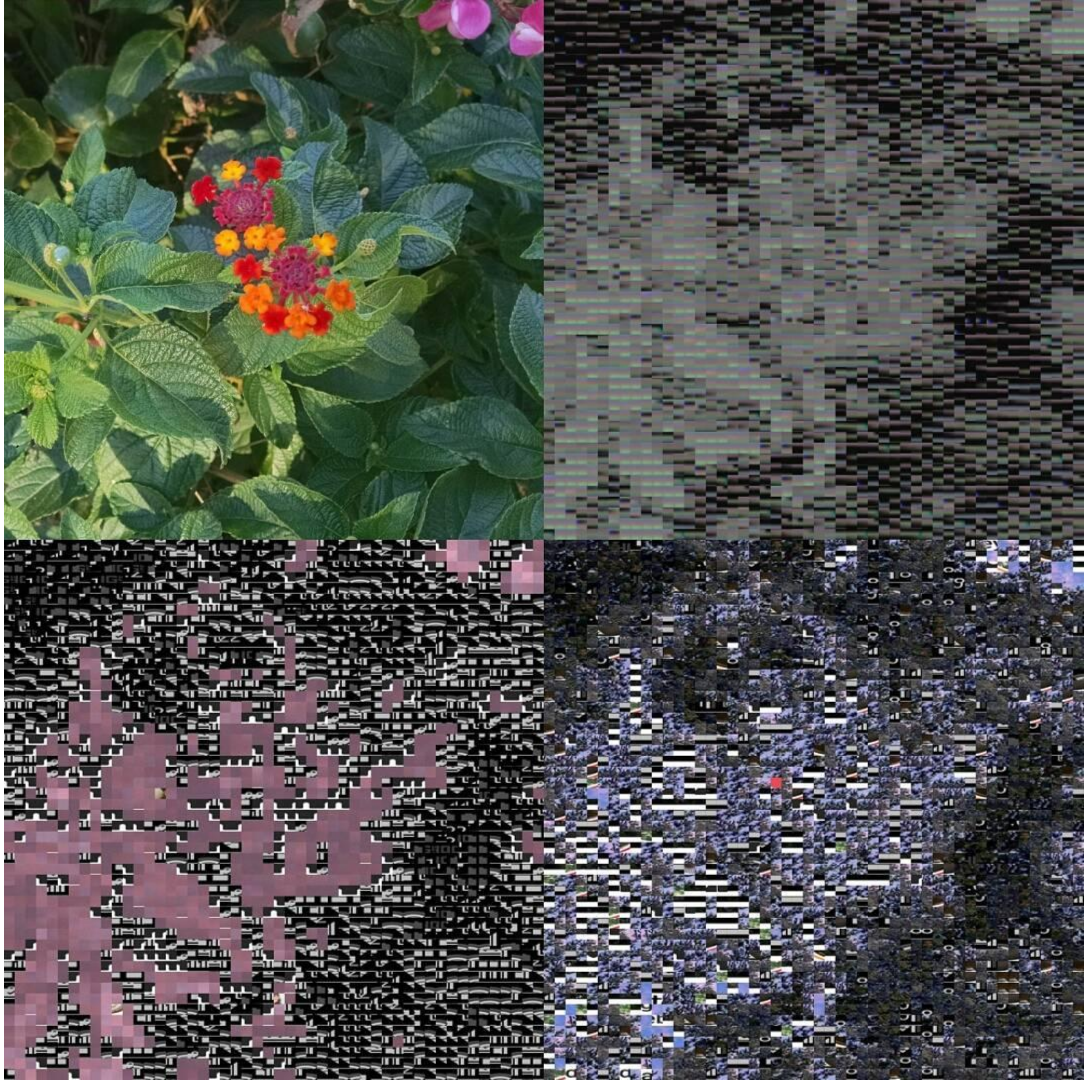
My Application

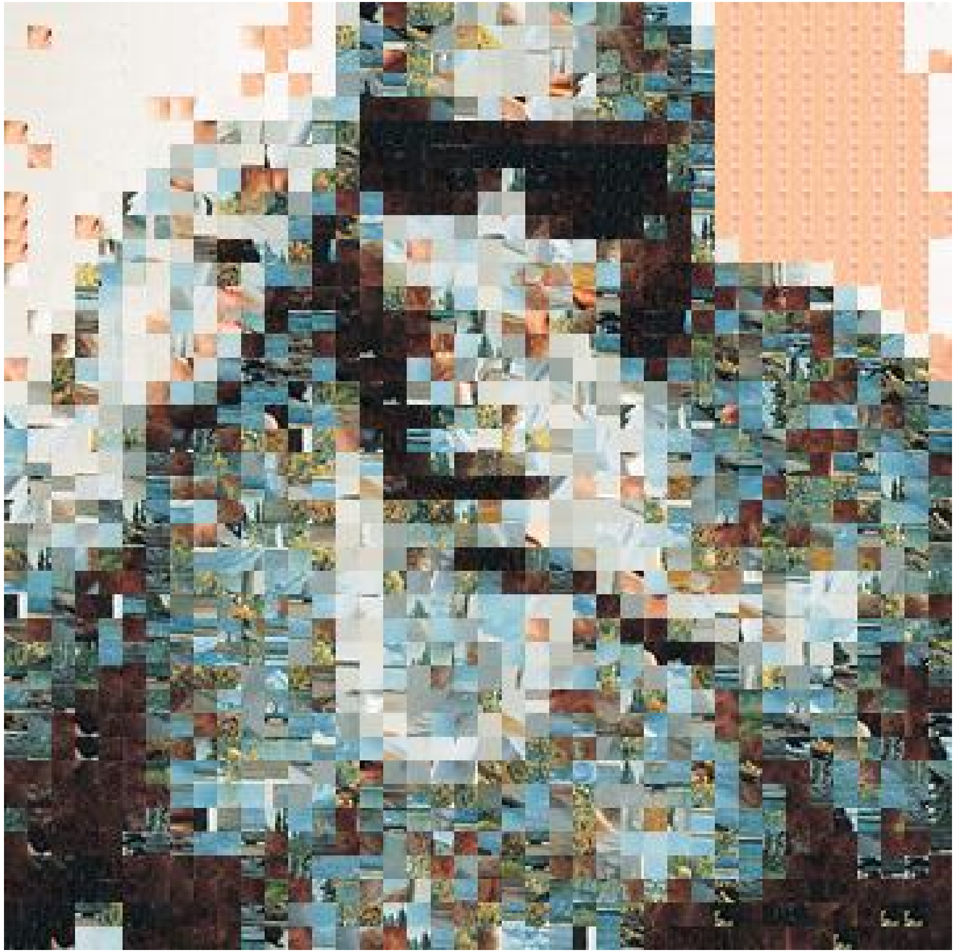
- Called “Flake”
  - Chops
  - Matches
- Examples

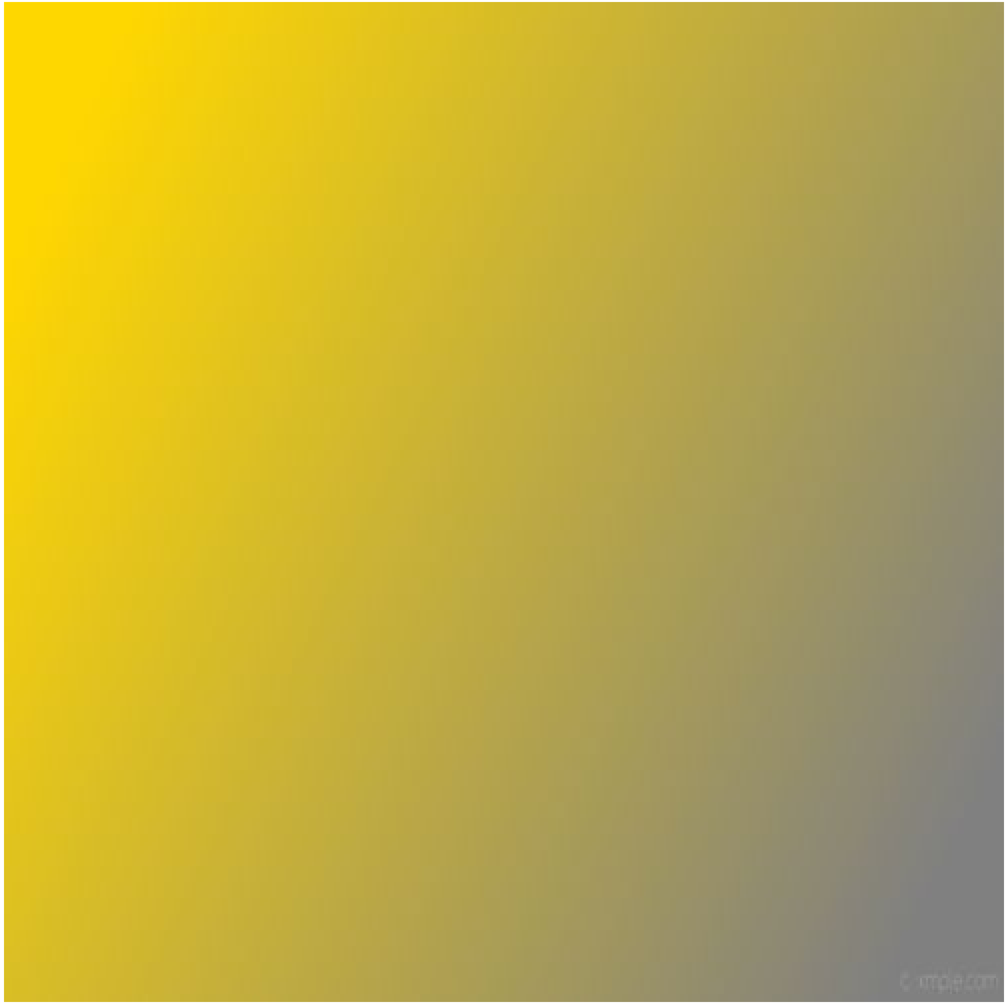


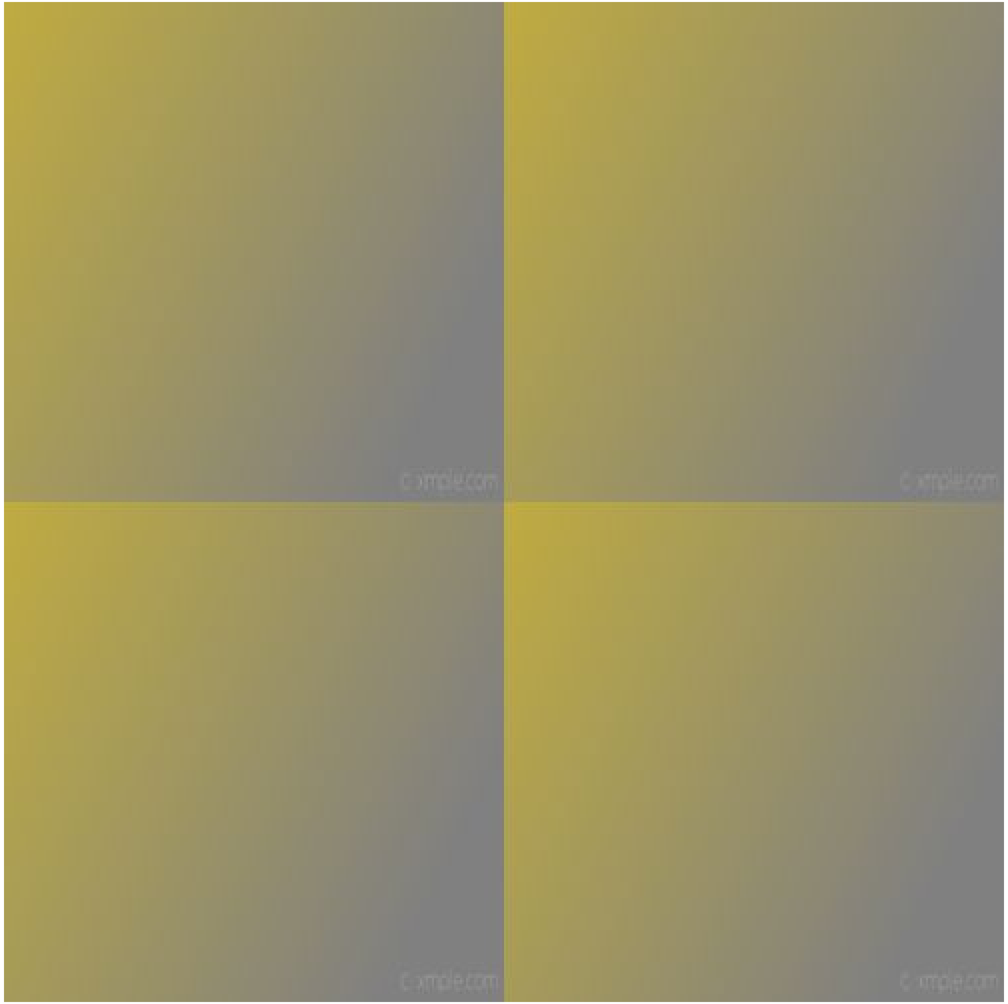


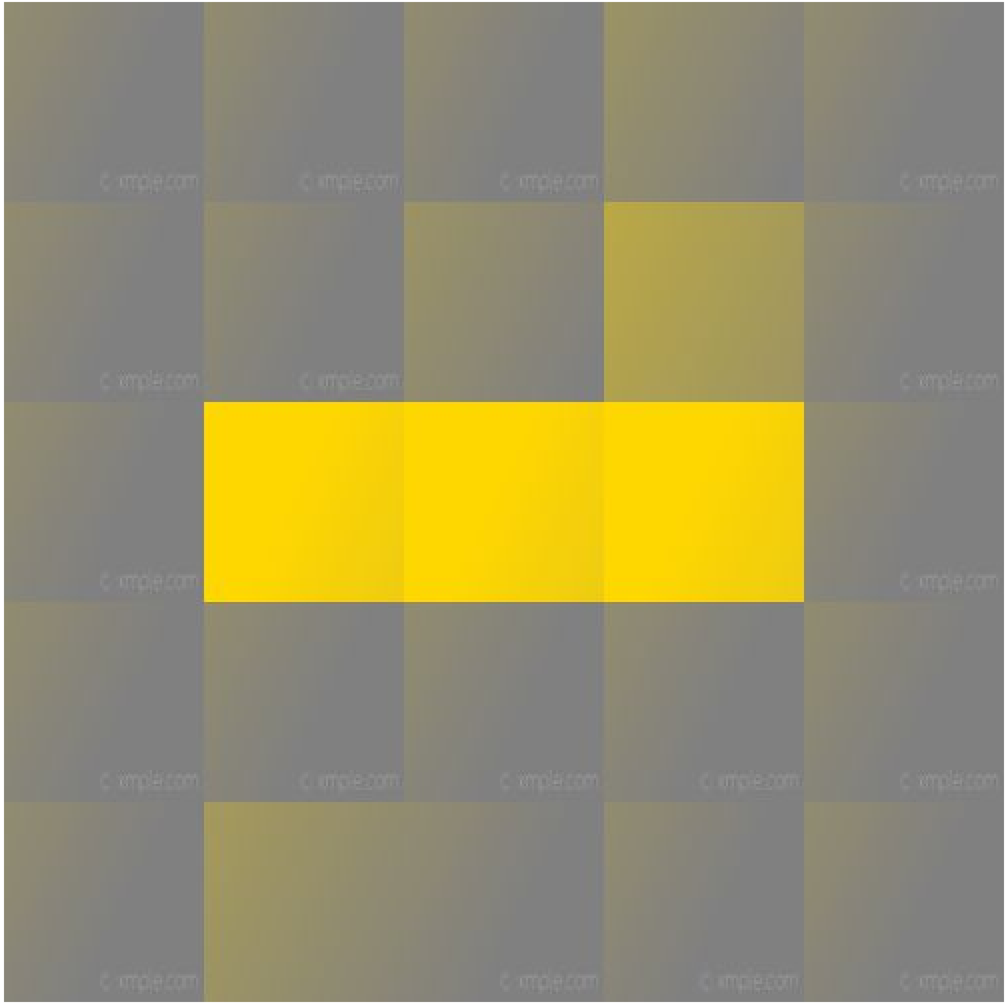




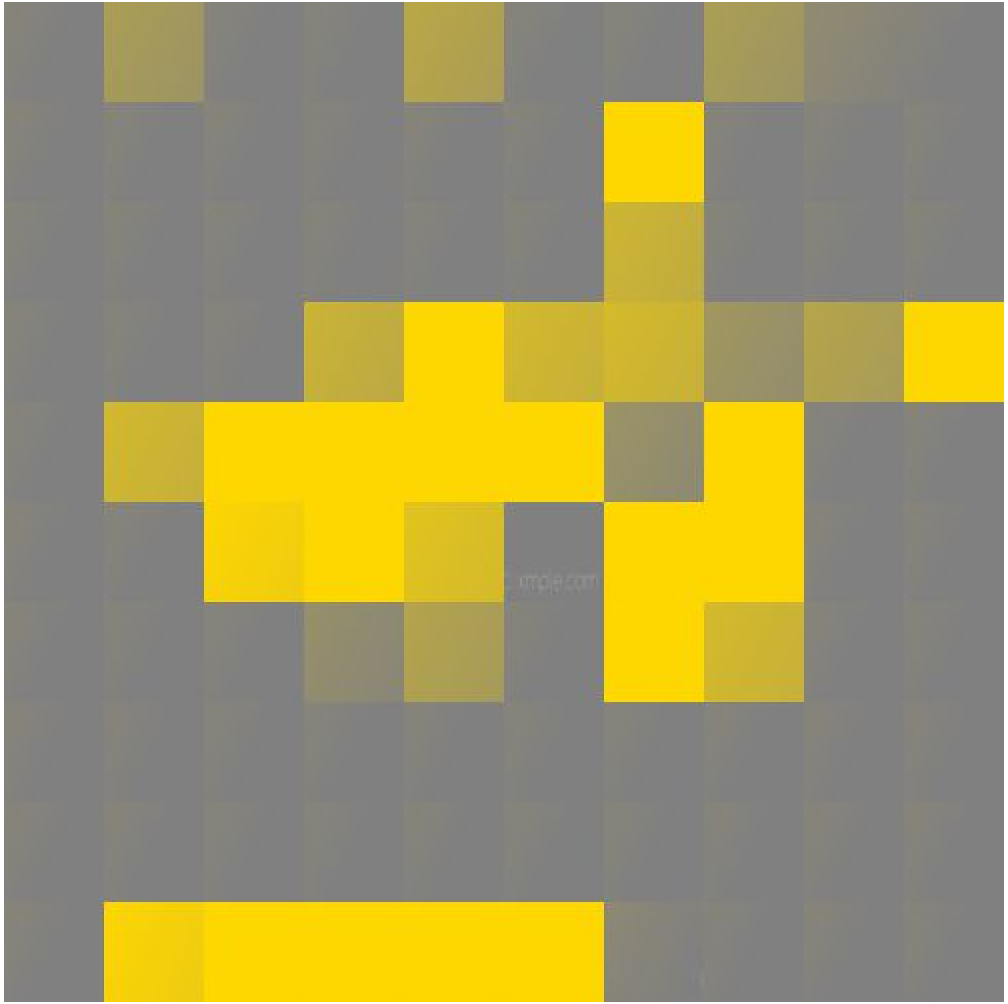


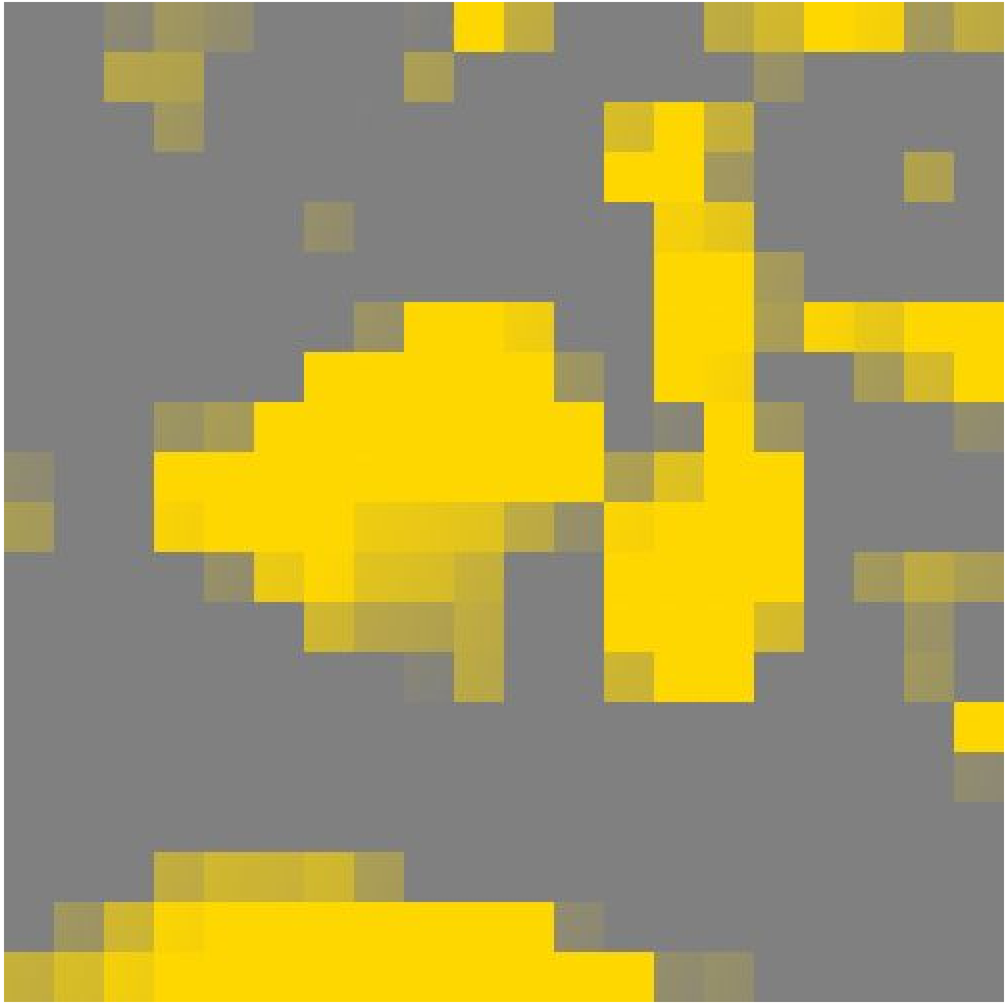


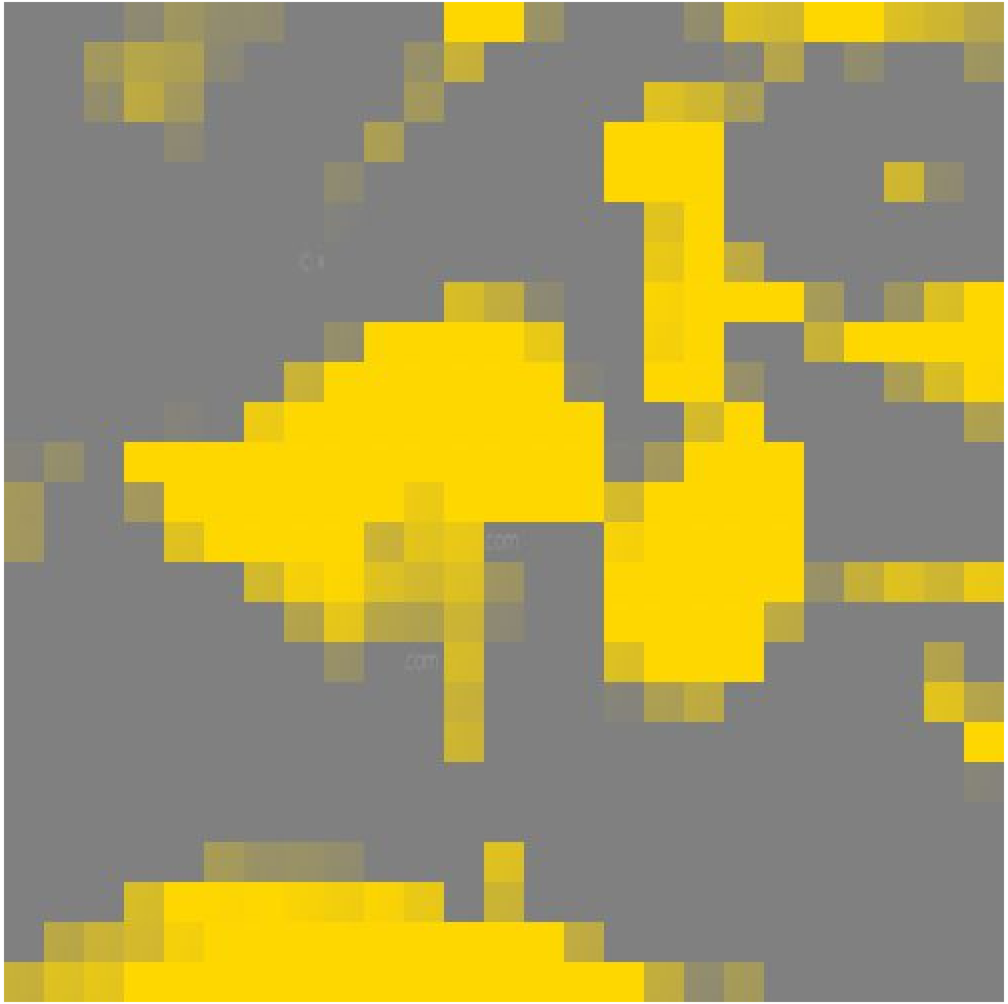


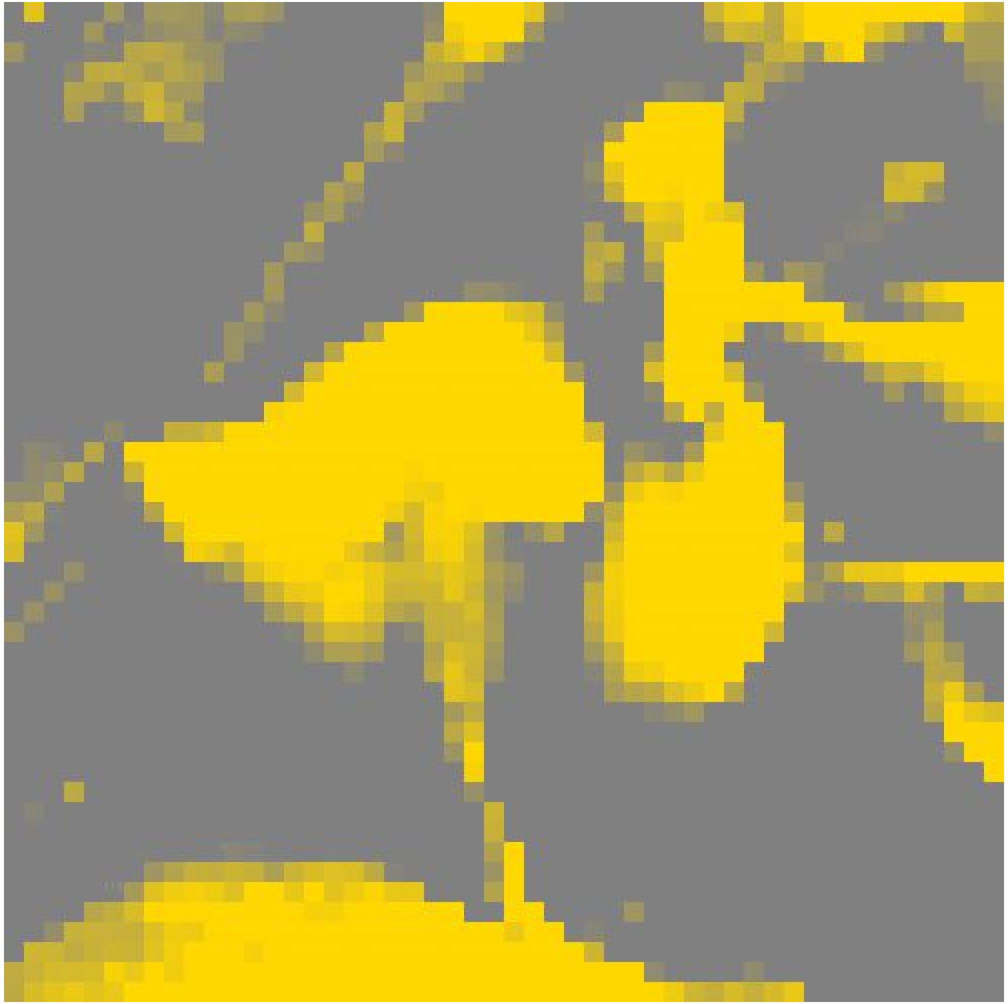


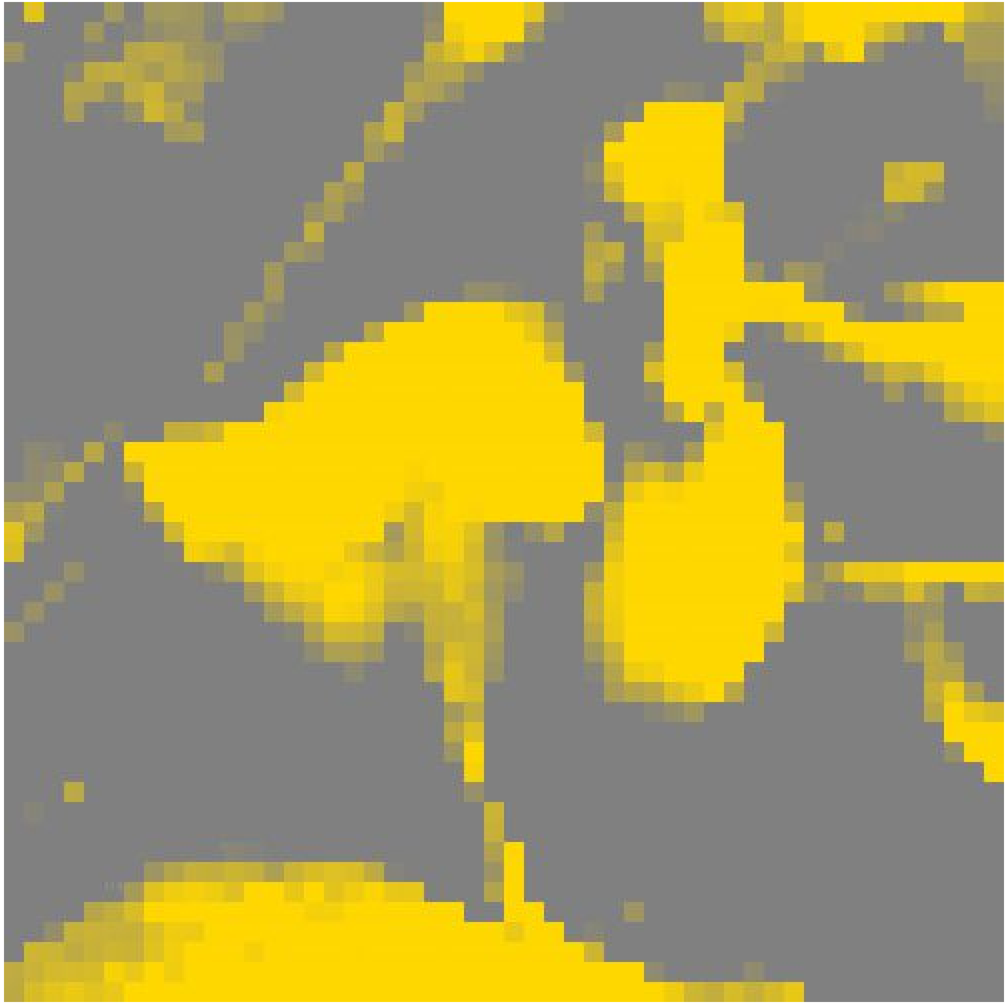
© xmple.com



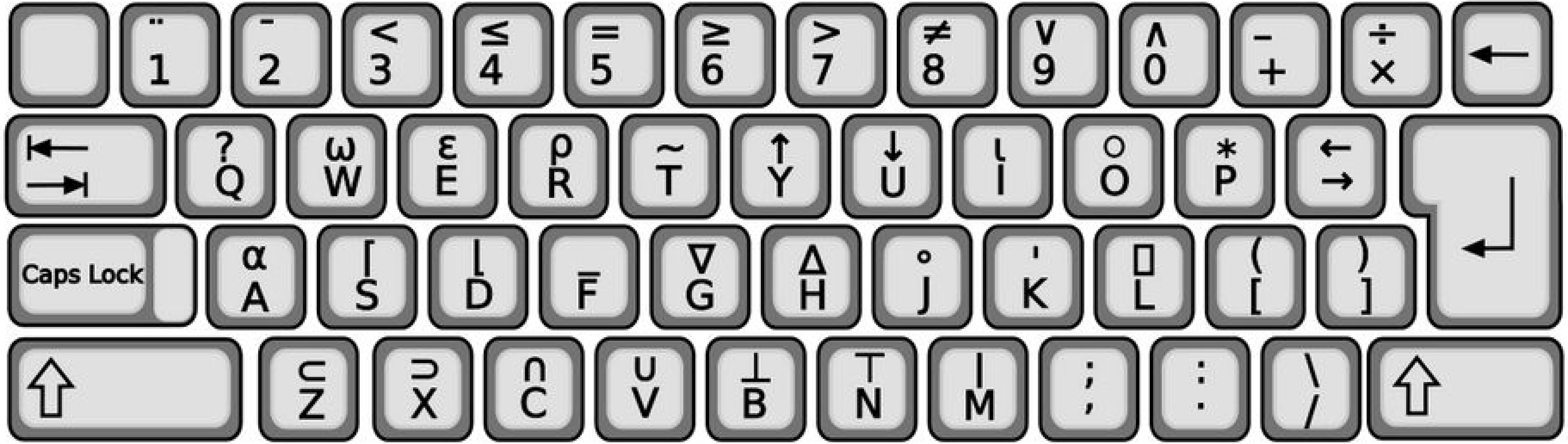








Currently online: YHNMJUIK.COM



Start Developing!

Requirements

Requirements

- Image Code

Requirements

- Image Code
  - Into APL
  - Out of APL

Requirements

- Image Code
    - Into APL
    - Out of APL
- Web Code

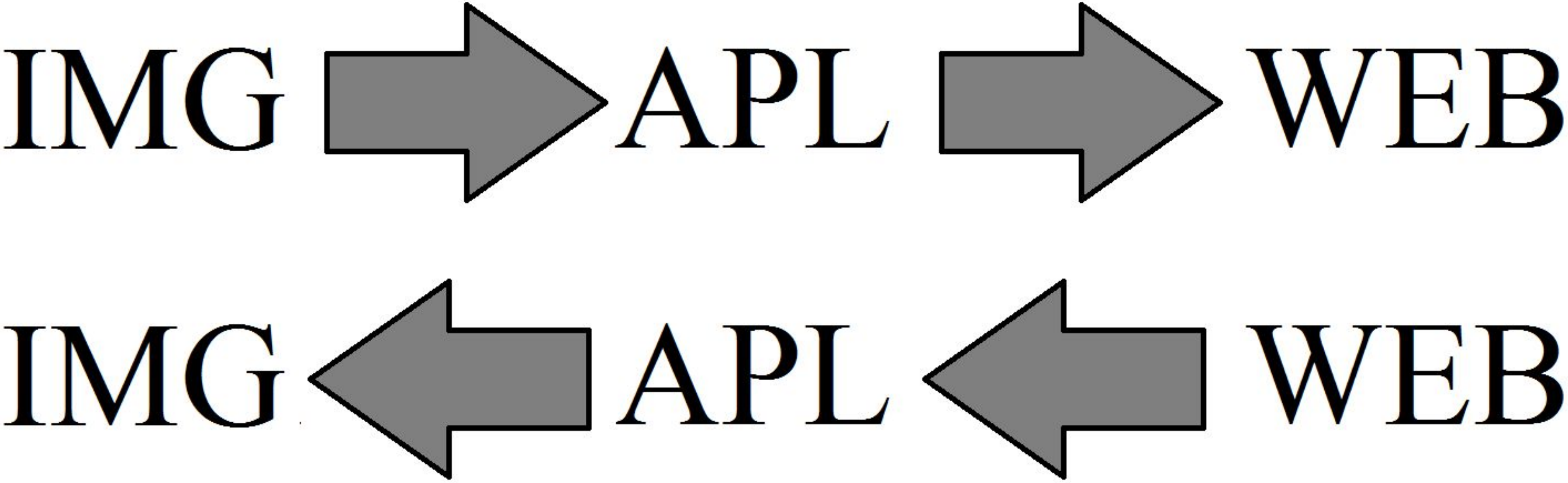
Requirements

- Image Code
  - Into APL
  - Out of APL
- Web Code
  - Output valid HTML
  - Parse form data as input

Requirements

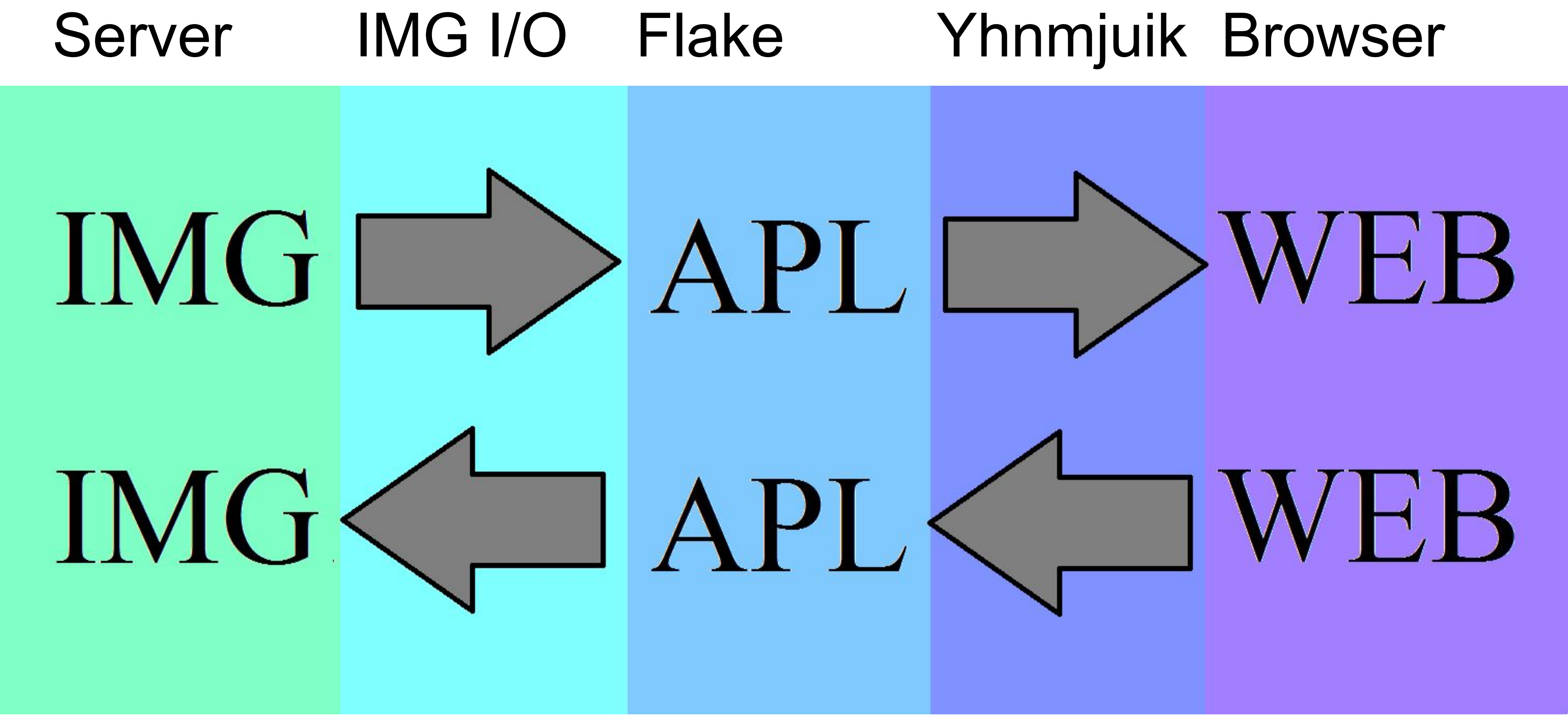
- Image Code
  - Into APL
  - Out of APL
- Web Code
  - Output valid HTML
  - Parse form data as input
- Flake Code

Requirements



IMG → APL → WEB

IMG ← APL ← WEB



Server IMG I/O Flake Yhnmjuik Browser

IMG APL WEB

IMG APL WEB

↑Those arrows are written in APL, also ↑

Architecture (Version 1)

Architecture (Version 1)

- PHP

Architecture (Version 1)

- PHP
  - Bash

Architecture (Version 1)

- PHP
  - Bash
    - J

Architecture (Version 1)

- PHP
    - Bash
        - J

A TOTAL MESS!

Architecture (Version 2)

Write every part in APL

Start Developing!

Start Developing!
(almost)

Set Up the Keyboard

Set Up the Keyboard

- This is not an APL problem

Set Up the Keyboard

- This is not an APL problem
    - This is an OS problem

Set Up the Keyboard

- This is not an APL problem
  - This is an OS problem
  - Experienced by a large number of APLers

Set Up the Keyboard

- This is not an APL problem
  - This is an OS problem
  - Experienced by a large number of APLers
- Unicode

Set Up the Keyboard

- This is not an APL problem
  - This is an OS problem
  - Experienced by a large number of APLers
- Unicode
  - The APL portion in “Miscellaneous Technical”

Set Up the Keyboard

```
sudo apt-get install subversion

sudo svn co http://svn.savannah.gnu.org/svn/apl/trunk

cd trunk

xmodmap support-files/Dyalog-Keyboard/apl.xmodmap-alexweiner
```

Set Up the Keyboard

- Files about the keyboard
    - trunk/README-3-keyboard
    - trunk/support-files/Dyalog-Keyboard/README

Start Developing

Start Developing!

(really!)

How to write a Bitmap Library

How to write a Bitmap Library

- Look at the spec

How to write a Bitmap Library

- Look at the spec
- Look at some C code

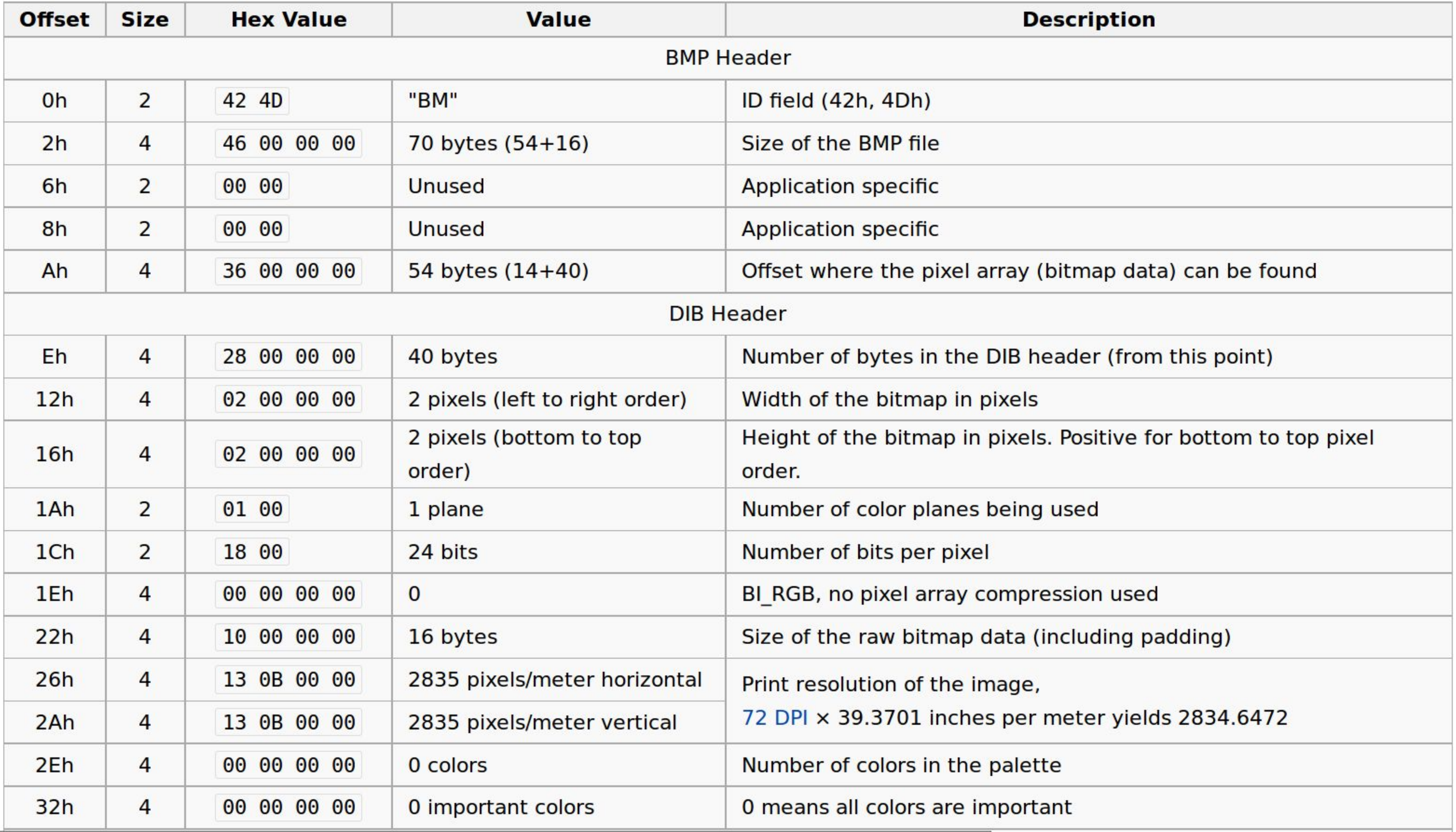
How to write a Bitmap Library

- Look at the spec
- Look at some C code
- The “Aha!” moment

Look at the Specification

Look at the Specification

- Wikipedia

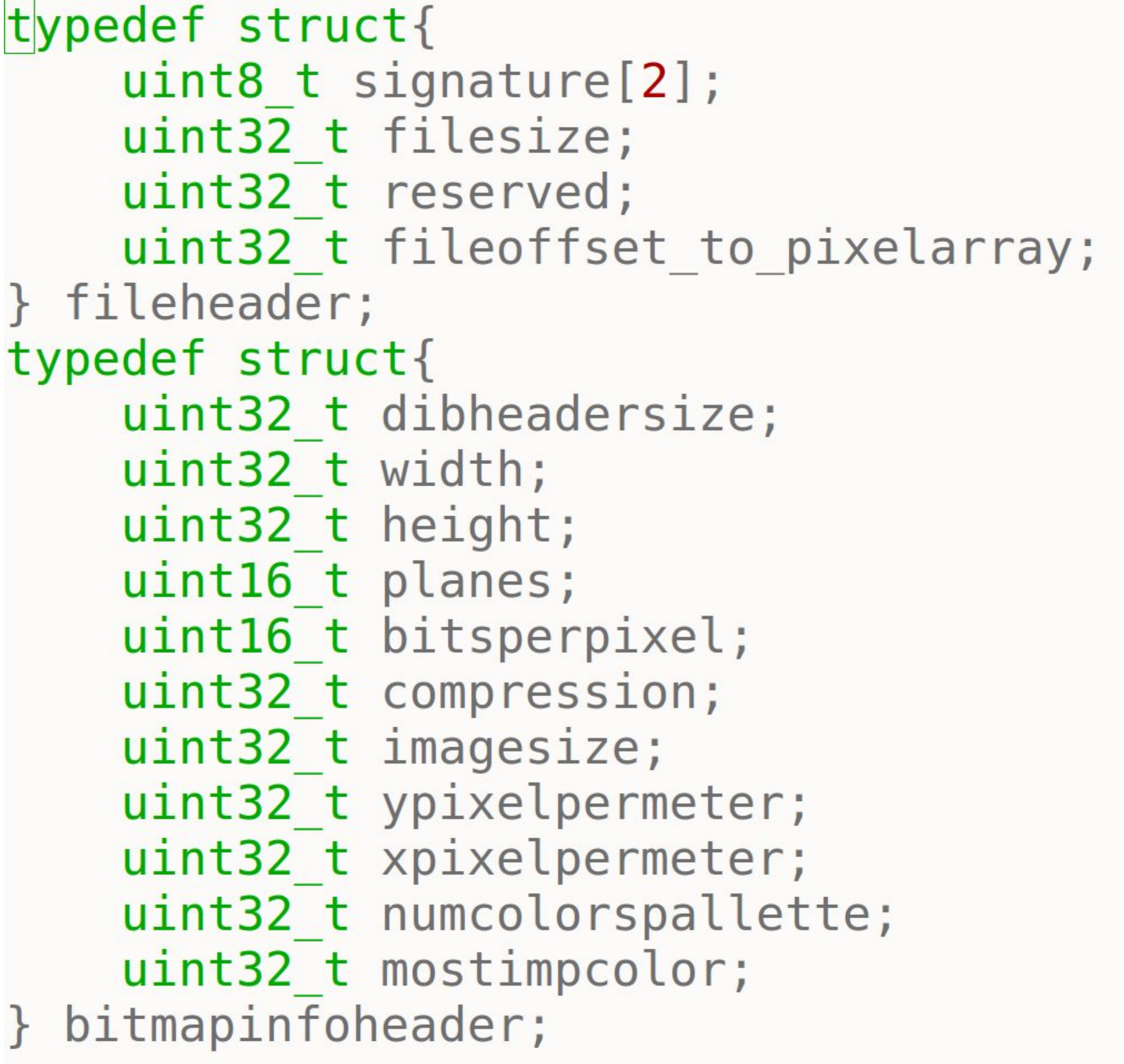


Offset	Size	Hex Value	Value	Description
BMP Header				
0h	2	42 4D	"BM"	ID field (42h, 4Dh)
2h	4	46 00 00 00	70 bytes (54+16)	Size of the BMP file
6h	2	00 00	Unused	Application specific
8h	2	00 00	Unused	Application specific
Ah	4	36 00 00 00	54 bytes (14+40)	Offset where the pixel array (bitmap data) can be found
DIB Header				
Eh	4	28 00 00 00	40 bytes	Number of bytes in the DIB header (from this point)
12h	4	02 00 00 00	2 pixels (left to right order)	Width of the bitmap in pixels
16h	4	02 00 00 00	2 pixels (bottom to top order)	Height of the bitmap in pixels. Positive for bottom to top pixel order.
1Ah	2	01 00	1 plane	Number of color planes being used
1Ch	2	18 00	24 bits	Number of bits per pixel
1Eh	4	00 00 00 00	0	BI_RGB, no pixel array compression used
22h	4	10 00 00 00	16 bytes	Size of the raw bitmap data (including padding)
26h	4	13 0B 00 00	2835 pixels/meter horizontal	Print resolution of the image, 72 DPI × 39.3701 inches per meter yields 2834.6472
2Ah	4	13 0B 00 00	2835 pixels/meter vertical	
2Eh	4	00 00 00 00	0 colors	Number of colors in the palette
32h	4	00 00 00 00	0 important colors	0 means all colors are important

BMP details

APL Representation	Actual Value	Verbal Description
46 0 0 0	70	Size of BMP file
36 0 0 0	54	Offset to pixel array
2 0 0 0	2	Image width
2 0 0 0	2	Image height

Look at C code



```c
typedef struct{
    uint8_t signature[2];
    uint32_t filesize;
    uint32_t reserved;
    uint32_t fileoffset_to_pixelarray;
} fileheader;
typedef struct{
    uint32_t dibheadersize;
    uint32_t width;
    uint32_t height;
    uint16_t planes;
    uint16_t bitsperpixel;
    uint32_t compression;
    uint32_t imagesize;
    uint32_t ypixelpermeter;
    uint32_t xpixelpermeter;
    uint32_t numcolorspallette;
    uint32_t mostimpcolor;
} bitmapinfoheader;
```

The “Aha!” Moment

- 32 bits are four 8-bit bytes

The “Aha!” Moment

```
256⊥¨⌽¨(46 0 0 0) (36 0 0 0) (2 0 0 0) (2 0 0 0)
```

```
46 36 2 2
```

How do you represent an image in APL?

How do you represent an image in APL?

- A 2D Matrix of vectors

How do you represent an image in APL?

- A 2D Matrix of vectors
  - Easier to look at while coding

How do you represent an image in APL?

- A 2D Matrix of vectors
  - Easier to look at while coding
  - Slow

How do you represent an image in APL?

- A 2D Matrix of vectors
  - Easier to look at while coding
  - Slow

```
content←offset↓bmp

partition←bytes_perpixel{ω⊂⍨∈

(α)/¨ια÷⍨ρω}contentbitmap←h w ρ partitioned
```

How do you represent an image in APL?

- A 3D Matrix

How do you represent an image in APL?

- A 3D Matrix
    - Faster code with less operations

How do you represent an image in APL?

- A 3D Matrix
  - Faster code with less operations
  - Extensible

How do you represent an image in APL?

- A 3D Matrix
  - Faster code with less operations
  - Extensible

```
bitmap←⍉width height dimension ⍴ content
```

APL and The Web

Passing web-form data with APL

- What is a web-form

Passing web-form data with APL

- What is a web-form
    - Syntax is HTML

Passing web-form data with APL

- What is a web-form
    - Syntax is HTML
    - HTTP methods:
        - GET
        - POST

Examples of GET and POST

```
<form action="code.apl"  method="get">
  <input type="submit" value="Click">
</form>



<form action="code.apl" enctype="multipart/form-data" method="post">
  <input type="file" name="name">
  <input type="submit" value="Click">
</form>
```

Passing form data with APL

- Language agnostic protocol

Passing form data with APL

- Language agnostic protocol
    - GET
        - “&” is the separator
        - environment variable called “QUERY_STRING”

Passing form data with APL

- Language agnostic protocol
  - POST
    - CRLF is the separator (Unicode 13 10)
    - Passed to stdin

```
Contents←{ω, FIOΔfread 0}⍣{α⊢FIOΔfeof 0}''
```

Demo

Demo

- Parsing a GET

Demo

- Parsing a GET

```
QUERY_STRING←‘var1=value1&var2=value2&var3=value3’

R←'&',QUERY_STRING

(+\R='&')⊂R
```

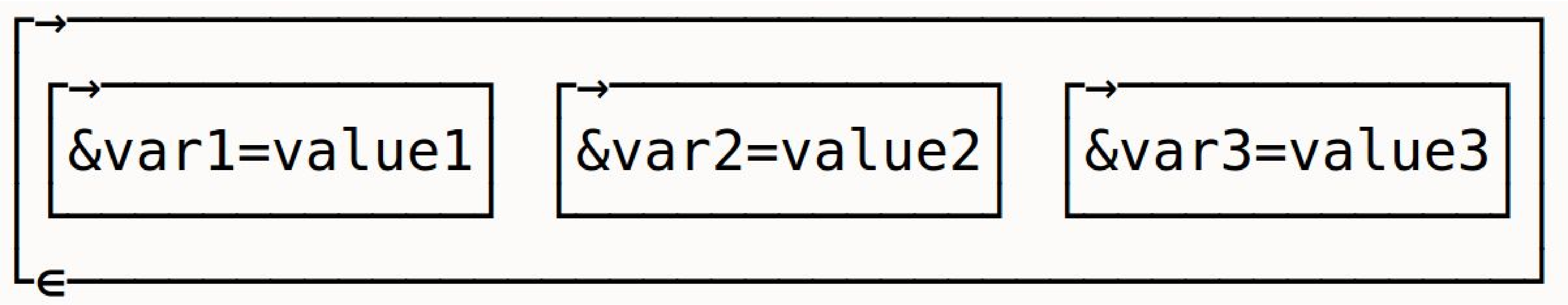
Demo

- Parsing a GET

  QUERY_STRING←‘var1=value1&var2=value2&var3=value3’

  R←'&',QUERY_STRING

  (+\R='&')⊂R



```
┌→─────────────────────────────────────────────────┐
│ ┌→────────────┐  ┌→────────────┐  ┌→────────────┐ │
│ │&var1=value1 │  │&var2=value2 │  │&var3=value3 │ │
│ └─────────────┘  └─────────────┘  └─────────────┘ │
└∊─────────────────────────────────────────────────┘
```

Other Goodies

Indexing

Indexing

- Good algorithms are Index-Origin independent

Indexing

- Good algorithms are Index-Origin independent
- How to convert from ⎕IO←0 to ⎕IO←1

```
ALEX[ 2 3 4 ]           ⍝ ⎕IO←1

ALEX[ 1 2 3 ]           ⍝ ⎕IO←0

ALEX[ ⎕IO + 1 2 3 ]     ⍝ Doesn’t matter
```

Control structures

- None are built in

Control structures

- None are built in
- Branching and line labels are included.

Control structures

```
LDI r16,0b00000001
CPI r16,0b00000001
BRNE somewhere

RJMP continue
somewhere:

continue:
```

```
r16←1
result←r16=1
→(result≠0)/somewhere

→continue
somewhere:

continue:
```

Questions?

Thank You!