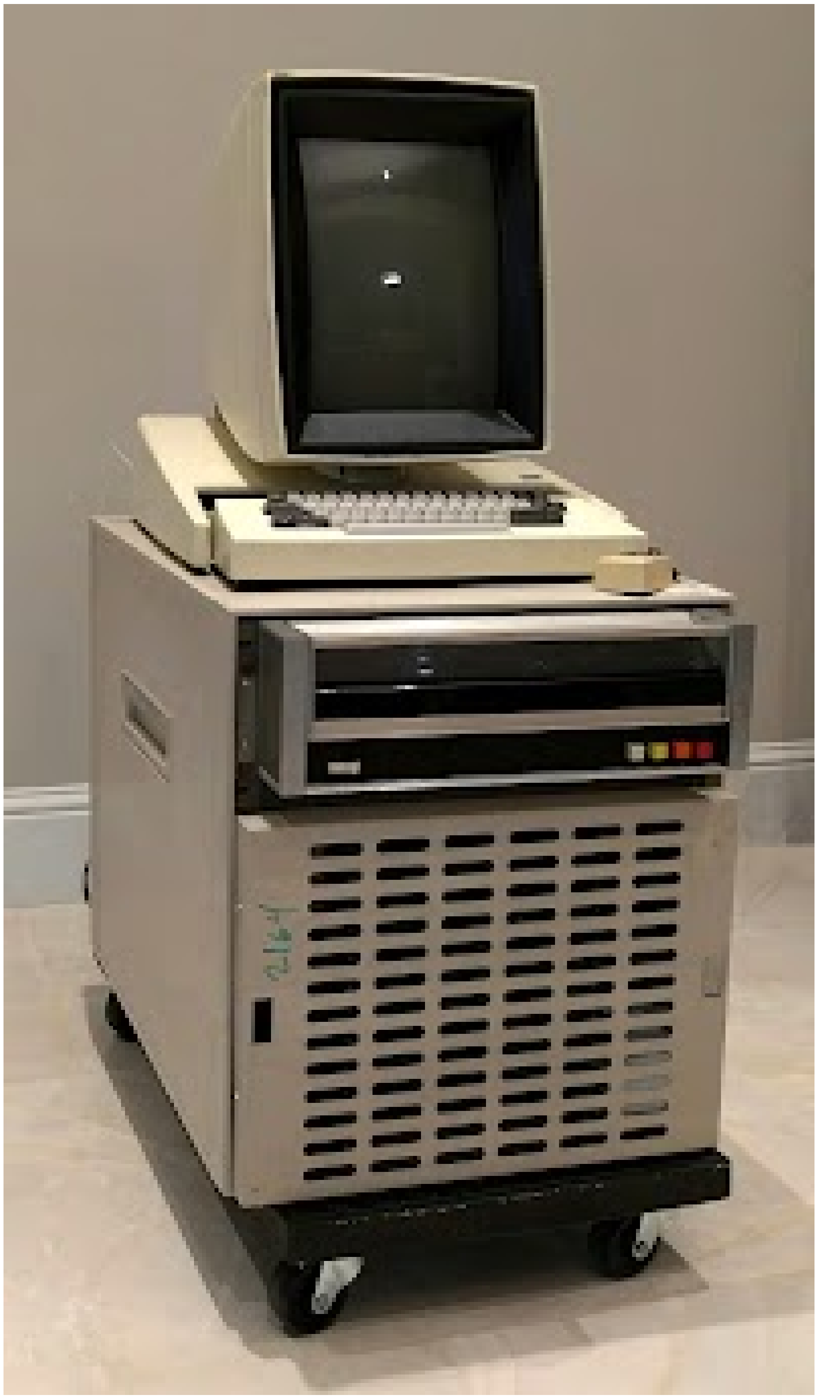
27 JUNE 2016 / RIGHTO.COM / 6 MIN READ / 105

"Hello world" in the BCPL language on the Xerox Alto simulator

The first programming language for the Xerox Alto was BCPL, the language that led to C. This article shows how to write a BCPL "Hello World" program using *Bravo*, the first WYSIWYG text editor, and run it on the Alto simulator.

The Xerox Alto is the legendary minicomputer from 1973 that helped set the direction for personal computing. Since I'm helping restore a Xerox Alto (details), I wanted to learn more about BCPL programming. (The influential Mesa and and Smalltalk languages were developed on the Alto, but those are a topic for another post.)



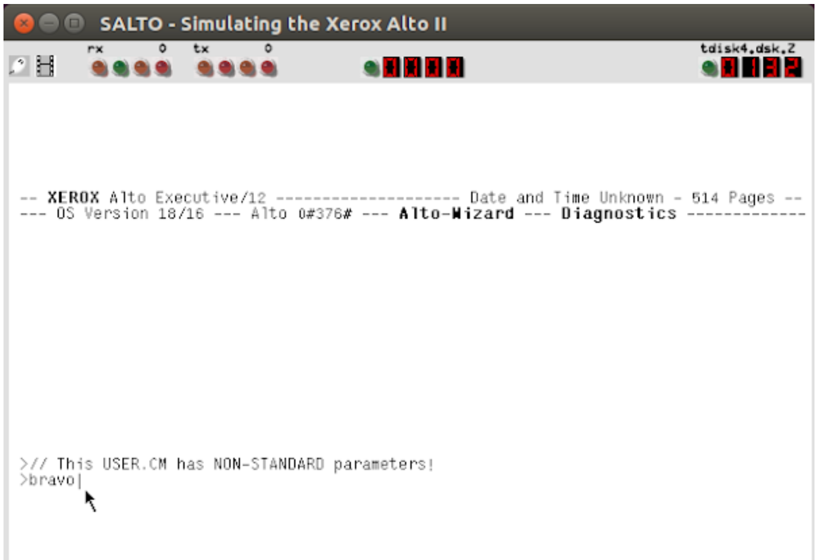
The Xerox Alto II XM computer. Note the video screen is arranged in portrait mode. Next to the keyboard is a mouse. The Diablo disk drive is below the keyboard. The base contains the circuit boards and power supplies.

Using the simulator

Since the Alto I'm restoring isn't running yet, I ran my BCPL program on *Salto*, an Alto simulator for Linux written by Juergen Buchmueller. To build it, download the simulator source from github.com/brainsqueezer/salto_simulator, install the dependencies listed in the README, and run `make`. Then run the simulator with the appropriate Alto disk image:

```
bin/salto disks/tdisk4.dsk.Z
```

Here's what the simulator looks like when it's running:



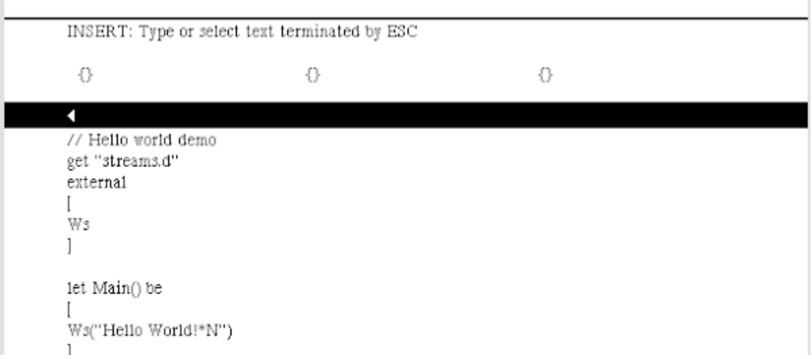
```
-- XEROX Alto Executive/12 -------------------- Date and Time Unknown - 514 Pages --
--- OS Version 18/16 --- Alto 0#376# --- Alto-Wizard --- Diagnostics -------------

>// This USER.CM has NON-STANDARD parameters!
>bravo|
```

The Salto simulator for the Xerox Alto.
(To keep this focused, I'm not going to describe everything you can run on the simulator, but I'll point out that pressing `?` at the command line will show the directory contents. Anything ending in `.run` is a program you can run, e.g. "pinball".)

Type `bravo` to start the Bravo text editor. Press `i` (for insert).
Enter the BCPL program:

```
// Hello world demo
get "streams.d"
external
[
Ws
]

let Main() be
[
Ws("Hello World!*N")
]
```

Here's a screenshot of the Bravo editor with the program entered:



INSERT: Type or select text terminated by ESC

{} {} {}

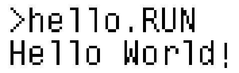
```
// Hello world demo
get "streams.d"
external
[
Ws
]

let Main() be
[
Ws("Hello World!*N")
]
```

A Xerox Alto 'Hello World' program for written in BCPL, in the Bravo editor.

Press `ESC` to exit insert mode.
Press `p` (put) to save the file.
Type `hello.bcpl` (the file name) and press `ESC` (not enter!).
Press `q` then `ENTER` to quit the editor.
Run the BCPL compiler, the linker, and the executable by entering the following commands at the prompt:

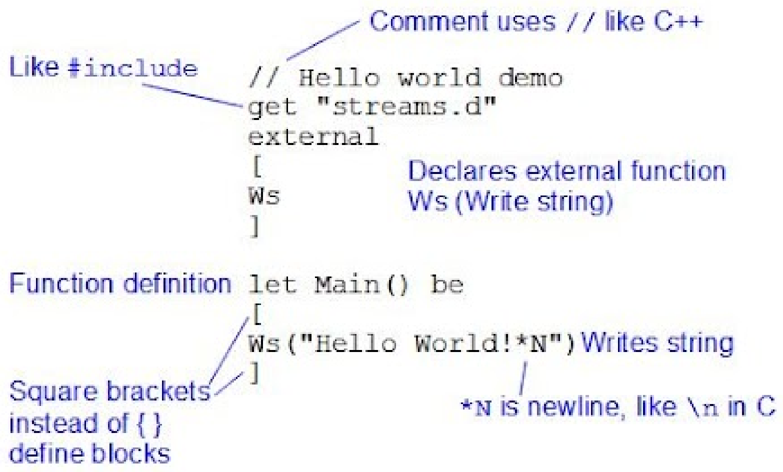
```
bcpl hello.bcpl
bldr/d/l/v hello
hello
```

If all goes well, the program will print "Hello World!" Congratulations, you've run a BCPL program.



```
>hello.RUN
Hello World!
```

Output of the Hello World program in BCPL on the Xerox Alto simulator.
The following figure explains the Hello World program. If you know C, the program should be comprehensible.



```
// Hello world demo
get "streams.d"
external
[
Ws
]
let Main() be
[
Ws("Hello World!*N")
]
```

'Hello World' program in BCPL with explanation.

The BCPL language

The BCPL language is interesting because it was the grandparent of C. BCPL (Basic Combined Programming Language) was developed in 1966. The B language was developed in 1969 as a stripped down version of BCPL by Ken Thompson and Dennis Ritchie. With the introduction of the PDP-11, system software needed multiple datatypes, resulting in the development of the C needed multiple datatypes, resulting in the development of the C language around 1972.

Overall, BCPL is like a primitive version of C with weirdly different syntax. The only type that BCPL supports is the 16-bit word, so it doesn't use type declarations. BCPL does support supports C-like structs and unions, including structs that can access bit fields from a word. (This is very useful for the low-level systems programming tasks that BCPL was designed for.) BCPL also has blocks and scoping rules like C, pointers along with lvalues and rvalues, and C-like looping constructs.

A BCPL program looks strange to a C programmer because many of the special characters are different and BCPL often uses words instead of special characters. Here are some differences:

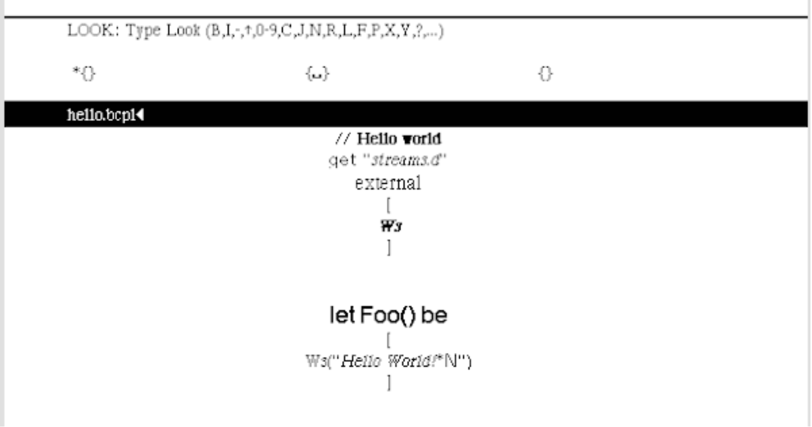
Blocks are defined with `[...]` rather than `{...}`.
For array indexing, BCPL uses `a!b` instead of `a[b]`.
BCPL uses `resultis 42` instead of `return 42`.
Semicolons are optional, kind of like JavaScript.
For pointers, BCPL uses `lv` and `rv` (lvalue and rvalue) instead of `&` and `*`. rvalues.
The BCPL operator `=>` (known as "heffalump"; I'm not making this up) is used for indirect structure references instead of C's `->`.
`selecton X into`, instead of C's `switch`, but cases are very similar with fall-throughs and default.
`lshift` and `rshift` instead of `<<` and `>>`.
`eq`, `ne`, `ls`, `le`, `gr`, `ge` in place of `==`, `!=`, `<`, `<=`, `>`, `>=`.
`test / ifso / ifnot` instead of `if / else`.

A BCPL reference manual is here if you want all the details of BCPL.

More about the Bravo editor

The Bravo editor was the first editor with WYSIWYG (what you see is what you get) editing. You could format text on the screen and print the text on a laser printer. Bravo was written by Butler Lampson and Charles Simonyi in 1974. Simonyi later moved to Microsoft, where he wrote Word, based on the ideas in Bravo. Steve Jobs saw the Alto when he famously toured Xerox Parc in 1979, and it inspired the GUI for the Lisa and Mac. However, Steve Jobs said in a commencement address, "[The Mac] was the first computer with beautiful typography. If I had never dropped in on that [calligraphy] course in college, the Mac would have never had multiple typefaces or proportionally spaced fonts. And since Windows just copied the Mac, it's likely that no personal computer would have them." This is absurd since the Alto had a variety of high-quality proportionally spaced fonts in 1974, before the Apple I was created, let alone the Macintosh.

The image below shows the Hello World program with multiple fonts and centering applied. Since the compiler ignores any formatting, the program runs as before. (Obviously styling is more formatting, the program runs as before. (Obviously styling is more useful for documents than code.)



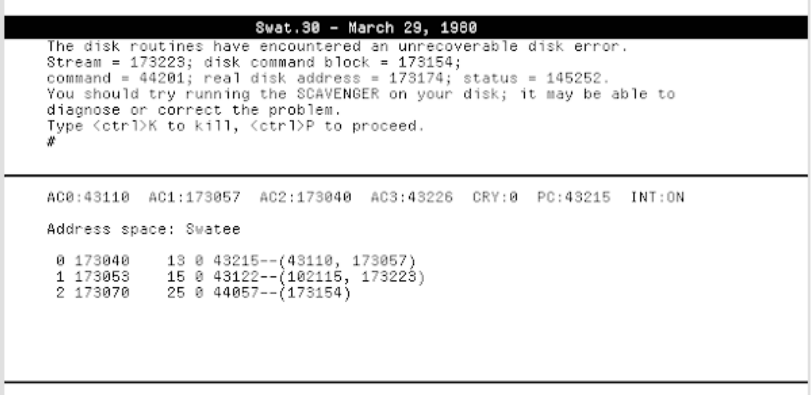
```
LOOK: Type Look (B,I,-,↑,0-9,C,J,N,R,L,F,P,X,Y,?,...)

*{}                    {␣}                    {}

hello.bcpl◀
// Hello world
get "streams.d"
external
[
Ws
]


let Foo() be
[
Ws("Hello World!*N")
]
```

The Bravo editor provides WYSIWYG formatting of text.
The manual for Bravo is here but I'll give a quick guide to Bravo if you want to try more editing. Bravo is a mouse-based editor, so use the mouse to select the text for editing. left click and right click under text to select it with an underline. The editor displays the current command at the top of the editing window. If you mistype a command, pressing `DEL` (delete) will usually get you out of it. Pressing `u` provides an undo.

To delete selected text, press `d`. To insert more text, press `i`, enter the text, then `ESC` to exit insert mode. To edit an existing file, start Bravo from the command line, press `g` (for get), then enter the filename and press `ESC`. To apply formatting, select characters, press `l` (look), and then enter a formatting code (0-9 to change font, b for bold, i for italics).

Troubleshooting

If your program has an error, compilation will fail with an error message. The messages don't make much sense, so try to avoid typos.
The simulator has a few bugs and tends to start failing after a few minutes with errors in the simulated disk. This will drop you into the Alto debugger, called *Swat*. At that point, you should restart the simulator. Unfortunately any files you created in the simulator will be lost when you exit the simulator.



```
                    Swat.30 - March 29, 1980
The disk routines have encountered an unrecoverable disk error.
Stream = 173223; disk command block = 173154;
command = 44201; real disk address = 173174; status = 145252.
You should try running the SCAVENGER on your disk; it may be able to
diagnose or correct the problem.
Type <ctrl>K to kill, <ctrl>P to proceed.
#

AC0:43110  AC1:173057  AC2:173040  AC3:43226  CRY:0  PC:43215  INT:ON

Address space: Swatee

 0 173040    13 0 43215--(43110, 173057)
 1 173053    15 0 43122--(102115, 173223)
 2 173070    25 0 44057--(173154)
```

If something goes wrong, you'll end up in Swat, the Xerox Alto's debugging system.

Conclusion

The BCPL language (which predates the Alto) had a huge influence on programming languages since it led to C (and thus C++, Java, C#, and so forth). The Bravo editor was the first WYSIWYG text editor and strongly influenced personal computer word processing. Using the Alto simulator, you can try both BCPL and Bravo for yourself by compiling a "Hello World" program, and experience a slice of 1970s computing history.

---

righto.com alto

Ken Shirriff
Read more posts by this author.

Read More

Latest Posts Vertex Pharmaceuticals: Building a Molecule Search Platform on AWS | Amazon Web Services |

> *I've finally learned what "upward compatible" means. It means we get to keep all our old mistakes.*
> *— Dennie van Tassel*

— LaptrinhX —

righto.com

Christmas shopping the IBM way: computerized gift selection in 1962

4 months ago — 3 min read

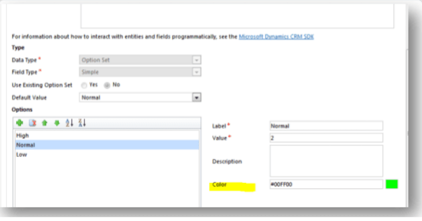
Reverse-engineering a tiny 1980s chip that plays Christmas tunes

4 months ago — 12 min read

Yamaha DX7 chip reverse-engineering, part 4: how algorithms are implemented

5 months ago — 20 min read

See all 214 posts →


For information about how to interact with entities and fields programmatically, see the Microsoft Dynamics CRM SDK

Type

Data Type * Option Set

Field Type * Simple

Use Existing Option Set ○ Yes ● No

Default Value Normal

Options

High

Normal

Low

Label * Normal

Value * 2

Description

Color #00FF00 🟩

ledgeviewpartners.com

5 years ago

Custom Coloring Dynamics CRM Dashboards

In Microsoft Dynamics CRM 2016, CRM Administrators have the ability to color code option lists. This can be very helpful when doing lists for things like case priority and just another way to make your dashboards...

L Ledgeview Partners

1 MIN READ

vidyard.com

5 years ago

Legal Rights in Your Video Soundtrack — 5 Questions to Ask Yourself

Choosing music for your video can be a tough choice and what can make that choice even more challenging are the legal considerations that come along with it. Unfortunately, this can be a grey and muddy area in terms of understanding what is and isn't legal and fair regarding the rights to your video's soundtrack. My goal is to help clarify and perhaps simplify an area of video production that is riddled with confusion and...

J Jon Spenceley

4 MIN READ

Popular : righto.com

Now! A Computer Makes Christmas Gift Shopping Easier

SIMPSON'S IBM ELECTRONIC GIFT SELECTOR

Another Free Christmas Service At Simpson's Toronto

Puzzled about presents for anyone on your list? . . . come to Simpson's . . . make four little ticks on a card . . . the IBM 1401 computer does the rest. It selects from hundreds and

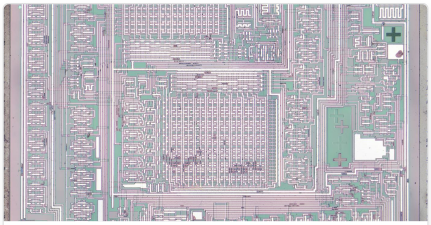
righto.com

4 months ago

Christmas shopping the IBM way: computerized gift selection in 1962

K Ken Shirriff

3 MIN READ



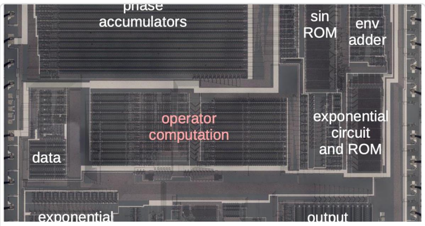
righto.com

4 months ago

Reverse-engineering a tiny 1980s chip that plays Christmas tunes

K Ken Shirriff

12 MIN READ



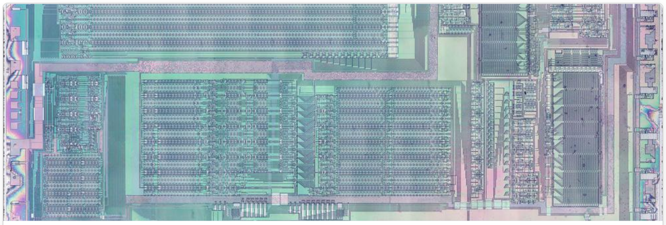
righto.com

5 months ago

Yamaha DX7 chip reverse-engineering, part 4: how algorithms are implemented

K Ken Shirriff

20 MIN READ



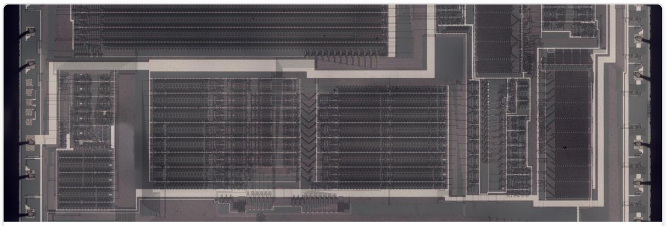
righto.com

5 months ago

Yamaha DX7 reverse-engineering, part III: Inside the log-sine ROM

K Ken Shirriff

8 MIN READ



righto.com

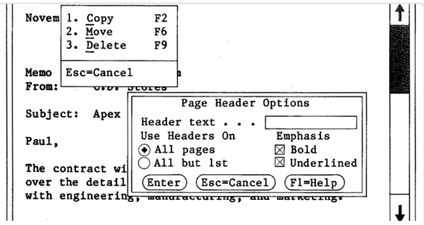
5 months ago

The Yamaha DX7 synthesizer's clever exponential circuit, reverse-engineered

K Ken Shirriff

16 MIN READ

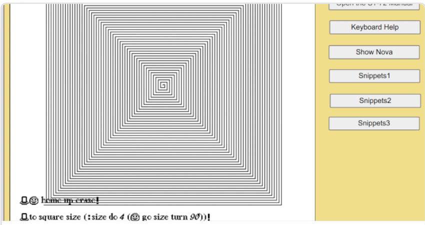
Recommended : alto


```
Novem 1. Copy     F2
      2. Move     F6
      3. Delete   F9

Memo  Esc=Cancel
From:
                 Page Header Options
Subject:  Apex  Header text . . .
                Use Headers On       Emphasis
Paul,           ◉ All pages          ☒ Bold
                ○ All but 1st        ☒ Underlined
The contract wi
over the detail  Enter  Esc=Cancel  F1=Help
with engineering, manufacturing, and marketing.
```

hackaday.com 1 year ago

The Origin of Cut, Copy, and Paste

A Al Williams 4 MIN READ



hackaday.com 1 year ago

Making a Little Smalltalk

A Al Williams 1 MIN READ

cordcuttersnews.com 1 year ago

TCL Announces 2020 Sound Bar Lineup

J Jess Barnes 1 MIN READ

github 1 year ago

Named Entities Recognition Annotator Tool for Europeana Newspapers

K KBNLresearch 3 MIN READ

Popular Tags @ LaptrinhX

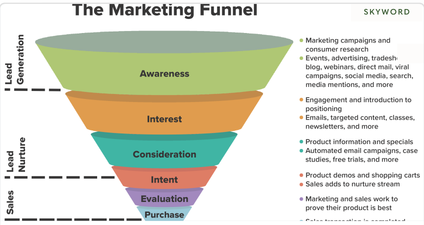
Most Viewed

LAPTRINHX

thomasthornton.cloud 1 year ago

Deploying Terraform from develop to production consecutively using Azure DevOps

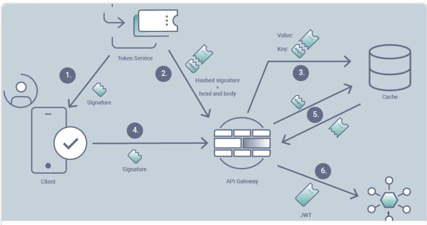
T Thomas Thornton 21 MIN READ


The Marketing Funnel
SKYWORD
Lead Generation
Lead Nurture
Sales
Awareness
Interest
Consideration
Intent
Evaluation
Purchase
Marketing campaigns and consumer research
Events, advertising, tradeshows, blog, webinars, direct mail, viral campaigns, social media, search, media mentions, and more
Engagement and introduction to positioning
Emails, targeted content, classes, newsletters, and more
Product information and specials
Automated email campaigns, case studies, free trials, and more
Product demos and shopping carts
Sales adds to nurture stream
Marketing and sales work to prove their product is best

skyword.com 1 year ago

How the Marketing Funnel Works From Top to Bottom

S Skyword Staff 7 MIN READ



nordicapis.com 1 year ago

Strategies for Integrating OAuth With API Gateways

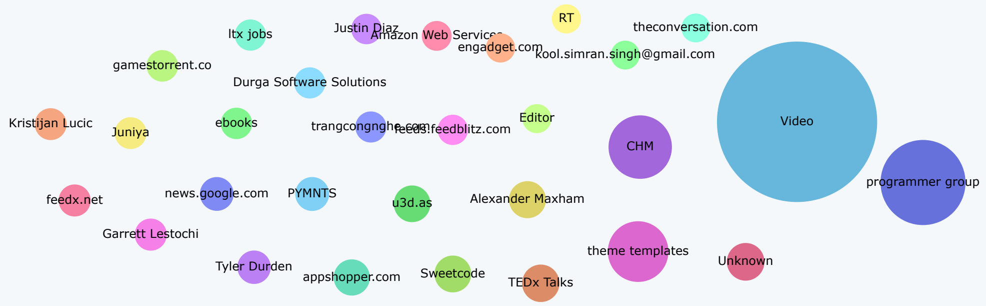
M Michał Trojanowski 5 MIN READ

tomsbiketrip.com 3 years ago

Terra Nova Starlite 2 Bikepacking Tunnel Tent: Long-Term Review & Photos

T Tom Allen 10 MIN READ

Popular Authors @ LaptrinhX



Last Post :

youtube.com

1 day ago

Google Play 政策更新重點解讀 - 2022 年 4 月 (中文講解)

A Android Developers

1 MIN READ

youtube.com

1 day ago

Liberty Mutual: Building a Serverless Call Center on AWS | Amazon Web Services

A Amazon Web Services

1 MIN READ

androidguys.com

1 day ago

Logitech Lift Vertical Ergonomic Mouse review

C Connor Fricke

5 MIN READ