How To Presentations 101

Peter Kos // 1/31/23 // WT

Before

During

After

Before


Idea

✨*Idea*

decide your audience

decide your audience

decide your audience

Intermediate Git

August 7th, 2022

decide your audience

people who used git
for a while, maybe are
lost?

decide your audience

people who used git
for a while, maybe are
lost?

people who know
enough to be
confused, frustrated

✨*Idea*

decide your audience

~~✨Idea~~

~~decide your audience~~

~~✨Idea~~

~~decide your audience~~

write an outline

```
outline!
    <screenshot of this>
    ideally, should be able to give talk w/o slides
        outline should be "well balanced"
    rule of 3
        <lotr reference>

(if code)
    make font big enough
        try to match what audience uses
            your psychadelic 8 space ident emacs config is cute but i can't read it
    make it DIGESTABLE
    give people time to read
    snippets are good if you can use them
    copy paste screen is great
    DESCRIBE THE WHY
        take people on a journey with you in your brain

slides
    BIG AND SIMPLE
    design for back of the room
        please, i cannot see 50% of presentations
    little to no text
        people will not listen to you, they will read
        get out of the template
        you do not need a title handholidng people through each slide
            if you make your talk interesting people will pay attention
    little to no code
        only what is expressly relevant
        no syntax highlighting needed
    little to no animations
    pick 3 colors, 2 fonts
        coolors.co
        fonts: fira sans, fira mono, recoleta, helvetica (NOT arial)
        color can be information, take cue from video games
    accessibility 101
        don't mix red/green (10%)
```

simple

hierarchical

rule of 3

```
outline!
    <screenshot of this>
    ideally, should be able to give talk w/o slides
        outline should be "well balanced"
    rule of 3
        <lotr reference>

(if code)
    make font big enough
        try to match what audience uses
            your psychadelic 8 space ident emacs config is cute but i ca
    make it DIGESTABLE
    give people time to read
    snippets are good if you can use them
    copy paste screen is great
    DESCRIBE THE WHY
        take people on a journey with you in your brain

slides
    BIG AND SIMPLE
    design for back of the room
        please, i cannot see 50% of presentations
    little to no text
        people will not listen to you, they will read
        get out of the template
        you do not need a title handholidng people through each slide
            if you make your talk interesting people will pay attention
    little to no code
        only what is expressly relevant
        no syntax highlighting needed
    little to no animations
    pick 3 colors, 2 fonts
        coolors.co
        fonts: fira sans, fira mono, recoleta, helvetica (NOT arial)
        color can be information, take cue from video games
    accessibility 101
        don't mix red/green (10%)
```

(ideally)

give the talk without slides

simple

hierarchical

rule of 3

(ideally)

give the talk without slides

(ideally)

give the talk without slides

~~✨Idea~~

~~decide your audience~~

write an outline

~~✨Idea~~

~~decide your audience~~

~~write an outline~~

write slides

write slides

write slides

live with code

make font big enough

make font big enough

try to match what
your audience uses

give time to read

}

And before that, in the pre-alpha days of Rust, arrays were defined with a variadic macro. The /* something */ above was a [T, ..$N], where T is the type, and ..$N defines a range (I believe -- old Rust is weird) up to the number of specified elements.

Ouch.

The standard library generates a new type for each 0..=N for type T (e.g., [T; 0], [T; 1], [T; 2]).

This means that if we want to implement anything on top of array -- Ord, PartialEq, etc. -- that means we need to implement it for all types of the array. (And indeed, in old versions of Rust, array docs were really messy, as they showed each implementation for all N!)

This problem is the perfect candidate for a new type of generic: const generics.

Const generics are presented very eloquently in RFC 2000: Const Generics. I'm going to summarize that RFC later on, with some tangents where appropriate, but let's start with a brief overview of the topic.

On its own, a const generic is generic that is restricted to be a specific constant value, specified (simply) with the const keyword2. I think they're best understood in the context of monomorphization.

[insert code]

This reveals the motivation behind the humble const generic. If we want to have a type that is exclusively distinguished by a constant (some might say "by association" of a constant), then a const generic is a fantastic qualifier. (Arrays are a good example here.) Otherwise, if a type will have many invocations with different values, it may be better to stick to a traditional parameter-in-struct approach.

Now that we've established the basics of const generics, let's dig more into

explain the why

write slides

(for real this time)

BIG

and

SIMPLE

design for the back of the room

make sure to put...

- *little to no text on the slides*
- *so that as a listener*
- *there is little to no text on each slide*
- *i can read each slide in a reasonable amount of time*
- *and also listen to the speaker*
- *instead of reading your powerpoint essay*

make sure to put
little to no text

(this is also accessibility!)

make sure to put

little to no text

look at this cat picture

make sure to put
little to no code

iOS

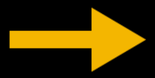
android

UIView: ...

UIButton: UIControl

MyButton: UIButton

What accessible views do you have?



```
class MyButtonUITest: XCUITest {
    func test_bgIsCorrectColor() {
        let button = view.descendants.matching(“MyButtonID”)
        XCTAssertEqual(button.bgColor, UIColor.blue)
    }
}
```

Protocols with associated types (or Self reqs) can now be used as types! Kinda.

```
protocol Connection { /* ... */ }

protocol Network {
  associatedtype ActiveConnection: Connection
  func initConnection(params: [String]) →
Connection
}
```

most of these have arguments, too:

```
alamborder
    --color/-c <color>
        A color name such as 'red', ‘blue'
    --width/-w <width>
        Desired width of border.
```

```
↝ alamborder -c “red” -w 2.0
```

make code

digestible

```
func scrollViewDidScroll(_ scrollView: UIScrollView) {
    let isTopOverscroll = contentOffset < (-1 * headerView.bounds.height)
    if lastContentOffset < contentOffset && !isTopOverscroll {
        scrollDirection = .down
    } else if lastContentOffset > contentOffset {
        scrollDirection = .up
    } else {
        scrollDirection = .unknown
    }
    lastContentOffset = scrollView.contentOffset.y
}
```

get rid of parts you don't need

```
func scrollViewDidScroll(_ scrollView: UIScrollView) {
    let isTopOverscroll = contentOffset < (-1 * headerView.bounds.height)
    if lastContentOffset < contentOffset && !isTopOverscroll {
        scrollDirection = .down
    } else if lastContentOffset > contentOffset {
        scrollDirection = .up
    } else {
        scrollDirection = .unknown
    }
    lastContentOffset = scrollView.contentOffset.y
}
```

get rid of parts you don’t need

```
func scrollViewDidScroll(_ scrollView: UIScrollView) {
    let isTopOverscroll = contentOffset < (-1 * headerView.bounds.height)
    if lastContentOffset < contentOffset && !isTopOverscroll {
        scrollDirection = .down
    } else if lastContentOffset > contentOffset {
        scrollDirection = .up
    } else {
        scrollDirection = .unknown
    }
    lastContentOffset = scrollView.contentOffset.y
}
```

get rid of parts you don’t need

```
if lastContentOffset < contentOffset {
    scrollDirection = .down
} else if lastContentOffset > contentOffset {
    scrollDirection = .up
} else {
    scrollDirection = .unknown
}
lastContentOffset = scrollView.contentOffset.y
```

reduce duplication

```
if lastContentOffset < contentOffset {
    scrollDirection = .down
} else if lastContentOffset > contentOffset {
    scrollDirection = .up
} else {
    scrollDirection = .unknown
}
lastContentOffset = scrollView.contentOffset.y
```

reduce duplication

```
if lastContentOffset < contentOffset {
    scrollDirection = .down
} else if lastContentOffset > contentOffset {
    scrollDirection = .up
} else {
    scrollDirection = .unknown
}
lastContentOffset = scrollView.contentOffset.y
```

reduce duplication

```
if lastContentOffset < contentOffset {
    // down
} else if lastContentOffset > contentOffset {
    // up
} else {
    // unknown
}
lastContentOffset = scrollView.contentOffset.y
```

don’t really syntax highlight

```
if lastContentOffset < contentOffset {
    // down
} else if lastContentOffset > contentOffset {
    // up
} else {
    // unknown
}
lastContentOffset = scrollView.contentOffset.y
```

don’t really syntax highlight

```
if lastContentOffset < contentOffset {
    // down
} else if lastContentOffset > contentOffset {
    // up
} else {
    // unknown
}
lastContentOffset = scrollView.contentOffset.y
```

don’t really syntax highlight

```
let isTopOverscroll = contentOffset < (-1 * headerView.bounds.height)
```

don’t really syntax highlight

```
let isTopOverscroll = contentOffset < (-1 * headerView.bounds.height)
if lastContentOffset < contentOffset {
    // down
} else if lastContentOffset > contentOffset {
    // up
} else {
    // unknown
}
lastContentOffset = scrollView.contentOffset.y
```

don’t really syntax highlight

```
let isTopOverscroll = contentOffset < (-1 * headerView.bounds.height)
if lastContentOffset < contentOffset && !isTopOverscroll {
    // down
} else if lastContentOffset > contentOffset {
    // up
} else {
    // unknown
}
lastContentOffset = scrollView.contentOffset.y
```

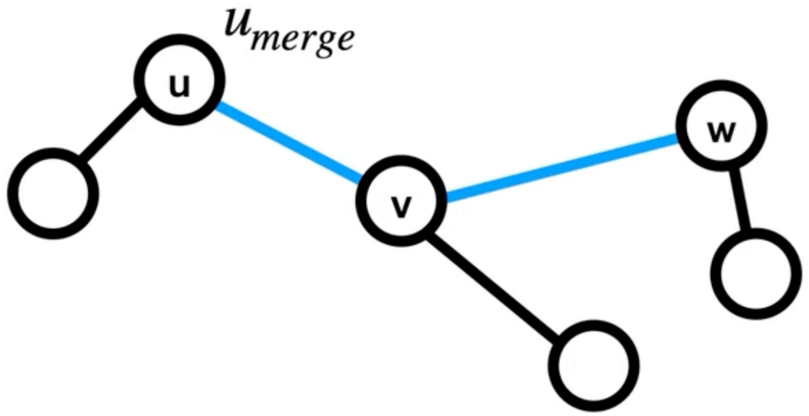
don’t really syntax highlight

```
let isTopOverscroll = contentOffset < (-1 * headerView.bounds.height)
if lastContentOffset < contentOffset && !isTopOverscroll {
    // down
} else if lastContentOffset > contentOffset {
    // up
} else {
    // unknown
}
lastContentOffset = scrollView.contentOffset.y
```

<light mode>

Algo 1

GraphToStar



Selection Merging **Pulling** Waiting Termination

Given $C(u) \leftrightarrow C(v)$ and $C(v) \leftrightarrow C(w)$

If leader of $C(v)$ did not activate in previous phase {
    $C(u)$ enters `merging` mode
} else if $C(u) \leftrightarrow C(v)$ and $C(v)$ is now empty {
    u activates uw
    $C(u)$ enters `merging` mode
} else {
    u activates uw, deactivates uv
    $C(u)$ remains in `pulling` mode
}

</light mode>

make sure to put
little to no animations

What happens when branches have *different history*?

file1.txt file2.txt

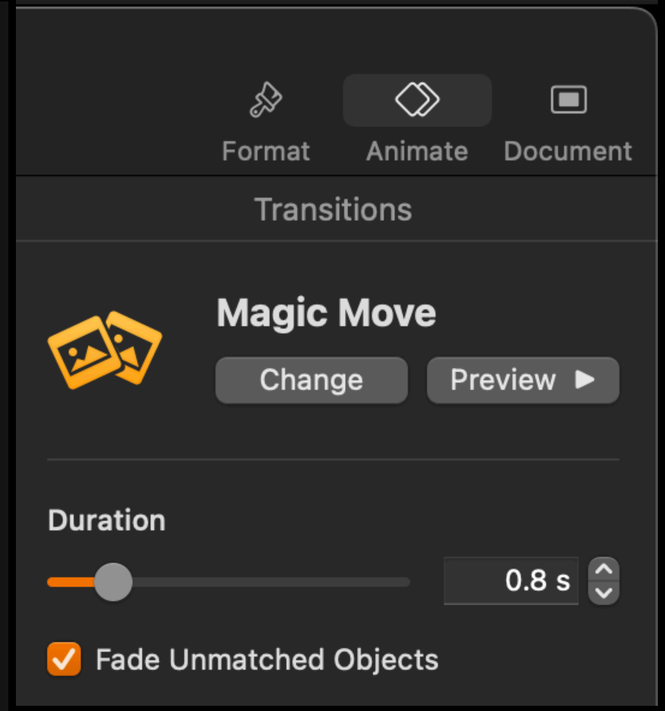
branch1

change file1.txt

create file3.txt

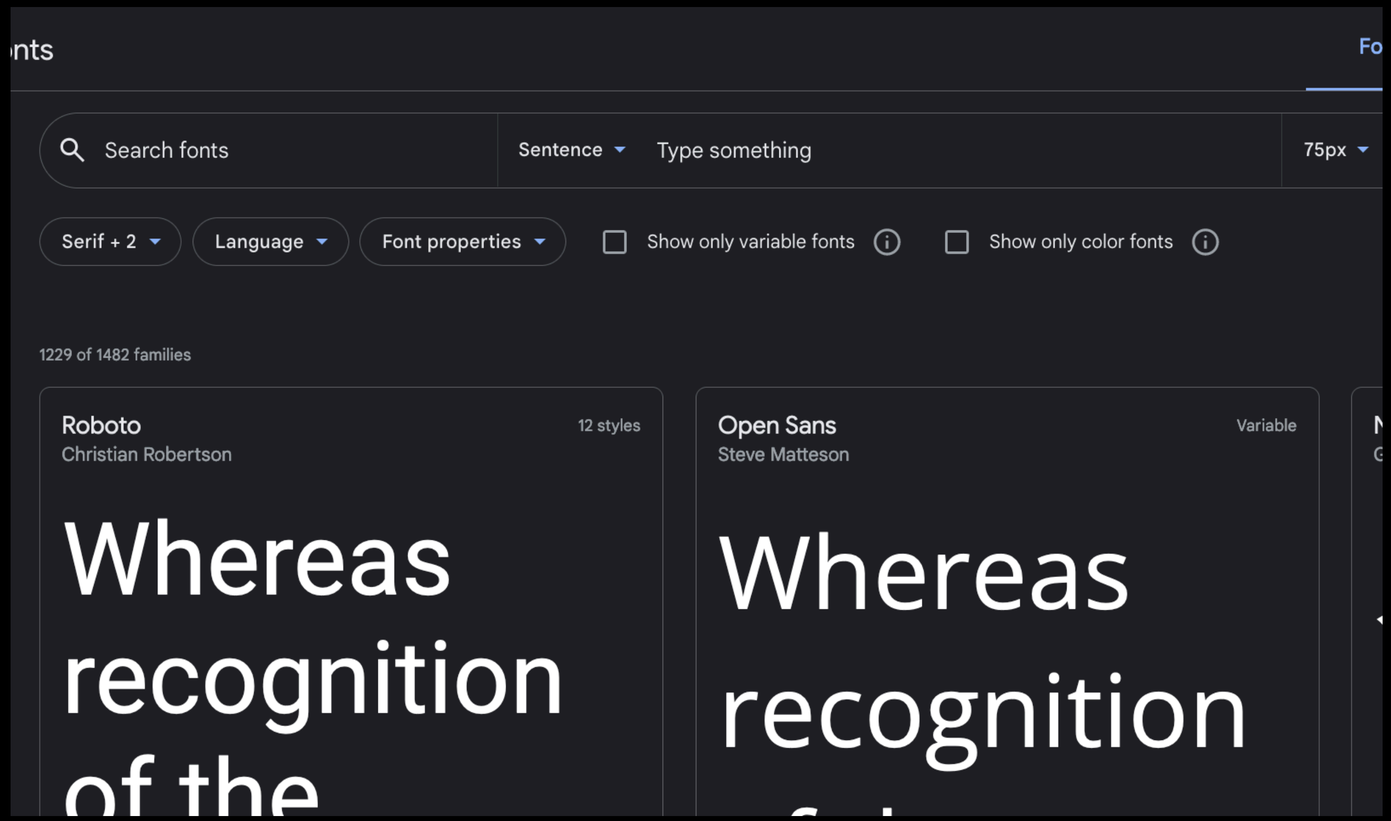
branch2

change file2.txt



pick some
colors and fonts

pick some colors and fonts



coolors.co

accessibility 101

accessibility 101

don't mix
red/green

(10% US men,
5% US women)

be careful about
blue/yellow

(1% overall)

contrast is 10x
worse on projector

~~✨Idea~~

~~decide your audience~~

~~write an outline~~

write slides

~~✨Idea~~

~~decide your audience~~

~~write an outline~~

~~write slides~~

~~✨Idea~~

~~decide your audience~~

~~write an outline~~

~~write slides~~

prep

prep

practise makes prefect

prep

pacing

prep

laptop, charger, usb-c adapter

prep

laptop, charger, usb-c adapter

backup .pdf slide copy DM'd in slack

prep

laptop, charger, usb-c adapter

backup .pdf slide copy DM'd in slack

comfortable outfit

water bottle

~~✨Idea~~

~~decide your audience~~

~~write an outline~~

~~write slides~~

prep

~~✨Idea~~

~~decide your audience~~

~~write an outline~~

~~write slides~~

~~prep~~

During

zoom *in person*

zoom

can you see my screen? 🥺👉👈

zoom

“Hey, DM me if there’s an issue!”

zoom

turn on do not disturb

(sponsored by peter's discord)

zoom

you don't have much
body language
to work with

zoom

you don't have much
body language
to work with

need to storytell
with your voice

in person

in person

(re: narrative)

your body language is your tool

in person

Not everyone is in person!

in person

Not everyone is in person!

but this is a good thing

recording

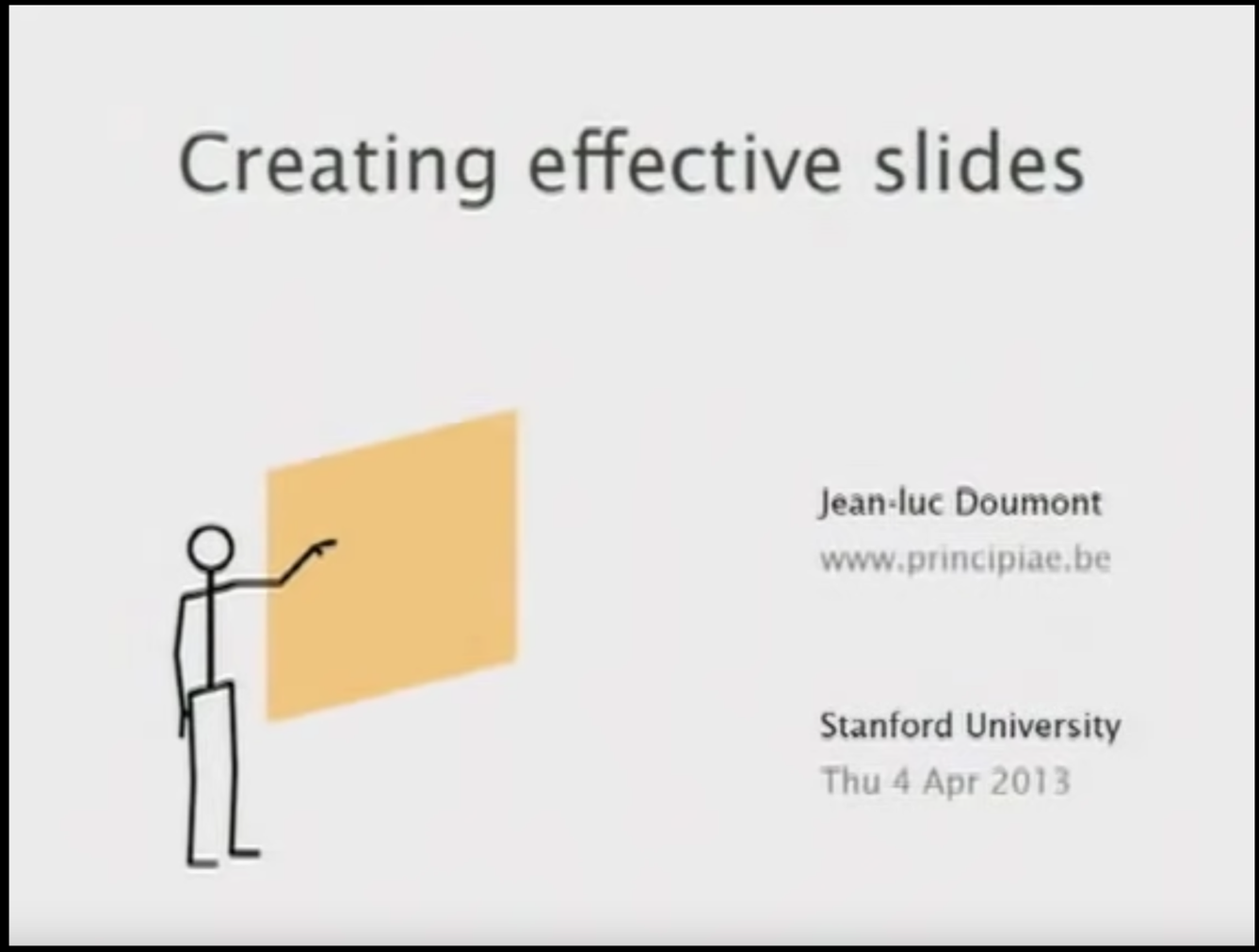
After

post slides

avenue for questions

Parting resources

Great for in-person, technical talks



A lot of today's info

Voice, narrative, education

How to Speak

IAP 2018

Patrick Henry Winston

MIT OCW

ocw.mit.edu

How To Presentations 101

Peter Kos // 1/31/23 // WT