ucraft [FRONTED MEETUPS 05]

StyledComponents

WHY, HOW AND ADVANCED TICKS

Complexity: medium
Library: React.js

WHO AM I?

SAYJEYHI
JAFAR REZAEI

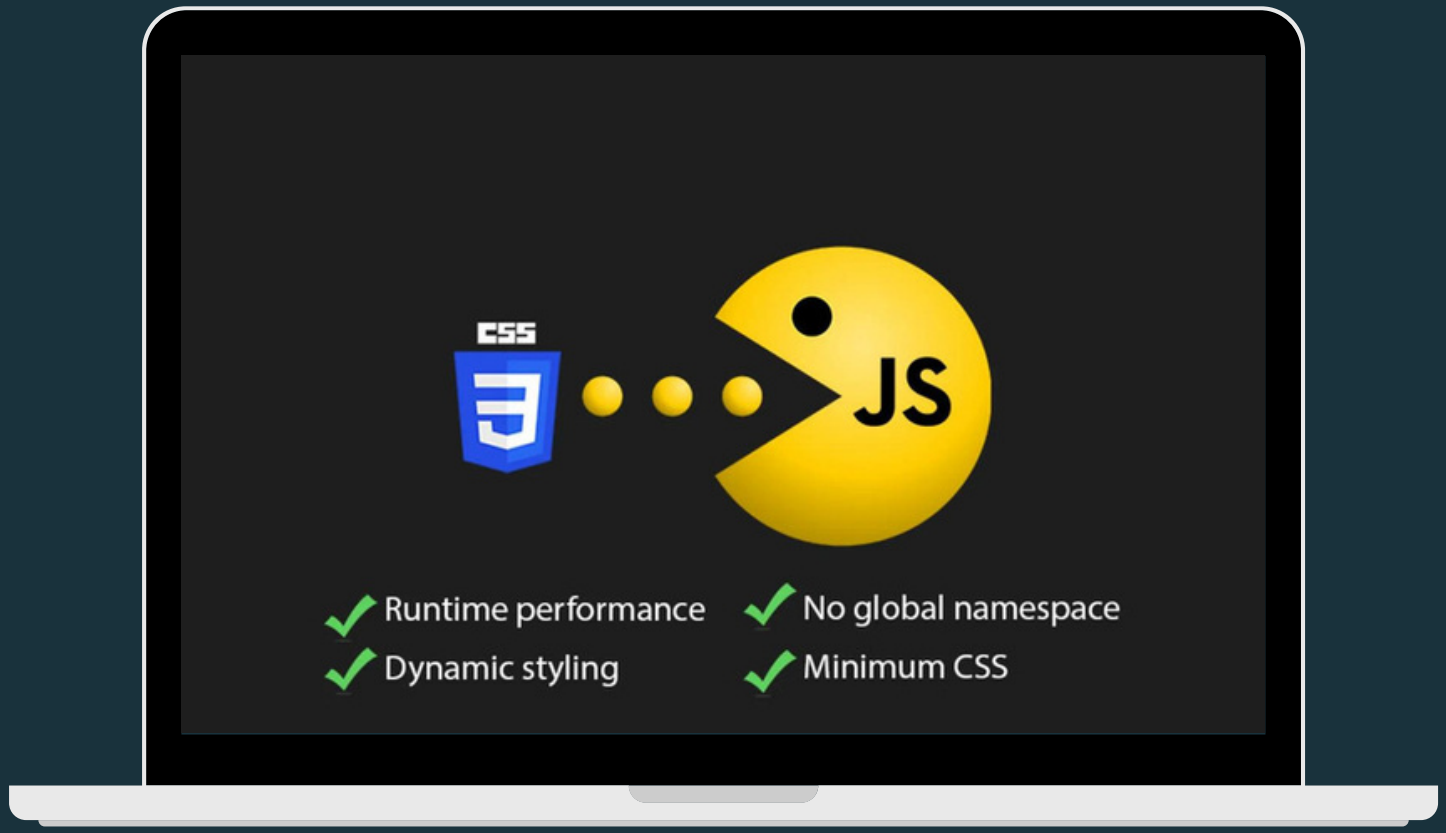
- writer of js fundamentals book in persian
- translator of reactjs QA book in persian
- working with HOORY team for 5 months
- did some open source contributions
- have 6+ years of experience with js
- got some first place at some competetions
- ...

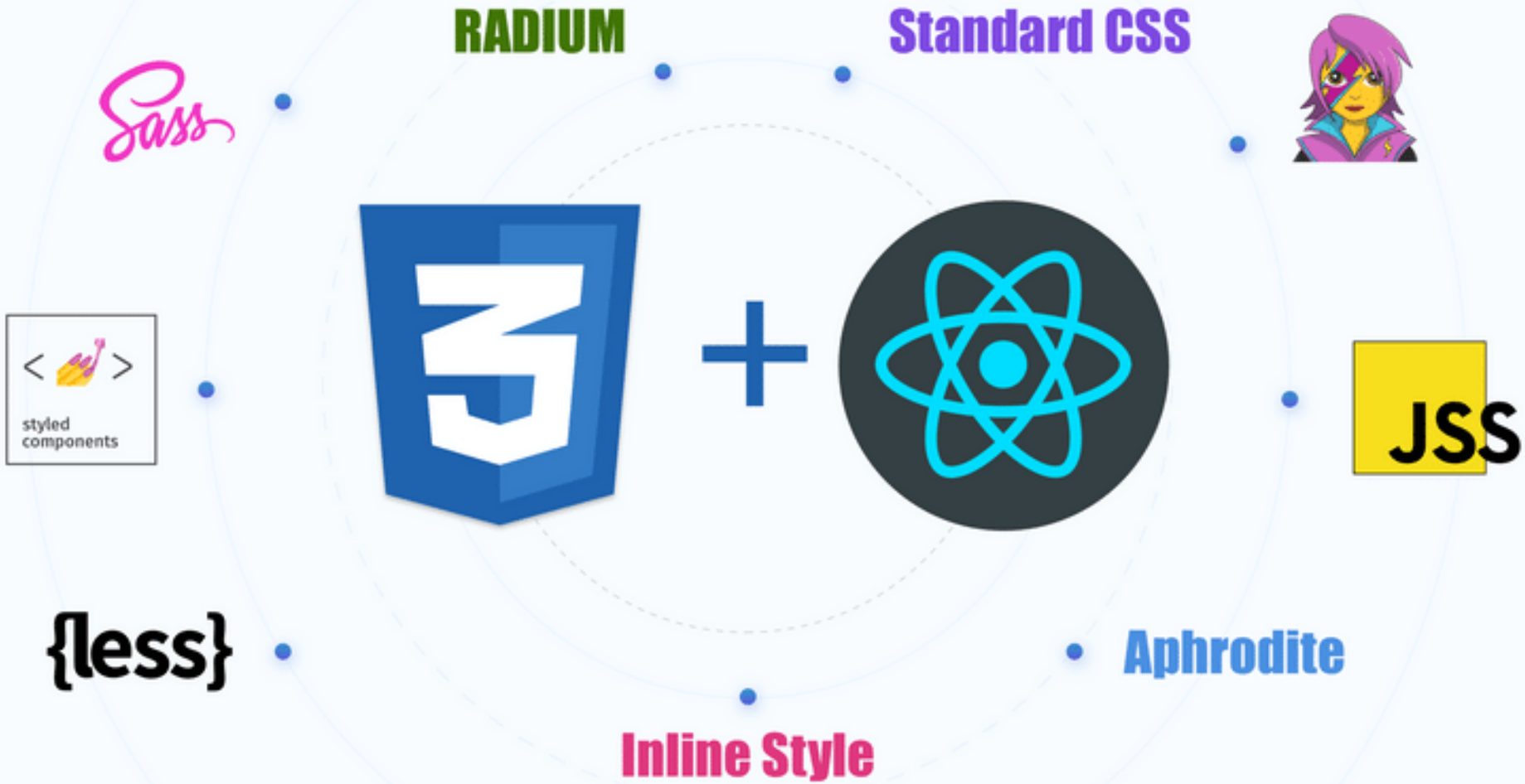
SAYJEYHI @ GMAIL.COM
WWW.SAYJEYHI.COM

STYLING

- OLD DAYS?
- NOWADAYS?

CSS-IN-JS PACMAN?



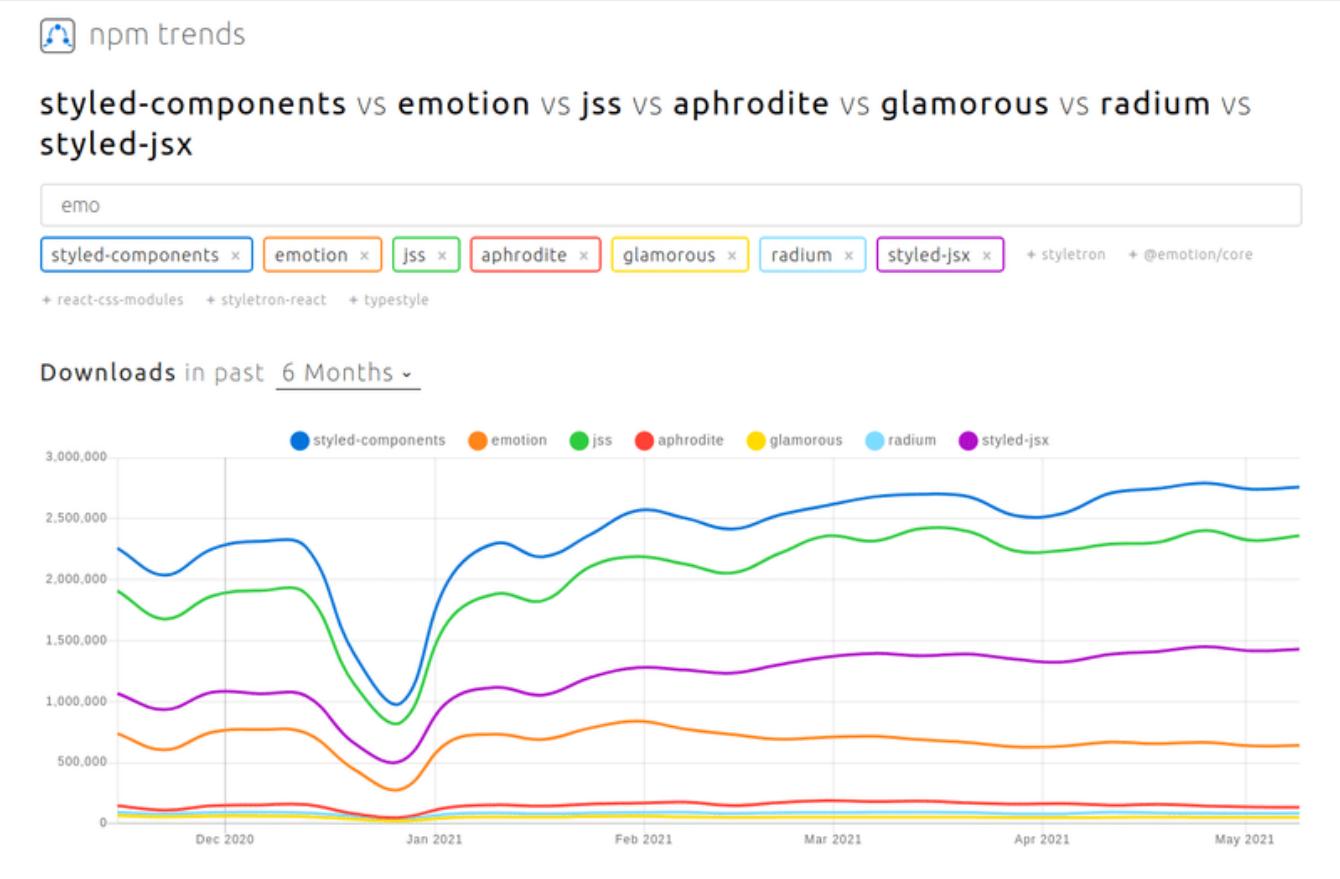

RADIUM
Standard CSS
Sass
styled components
JSS
{less}
Aphrodite
Inline Style

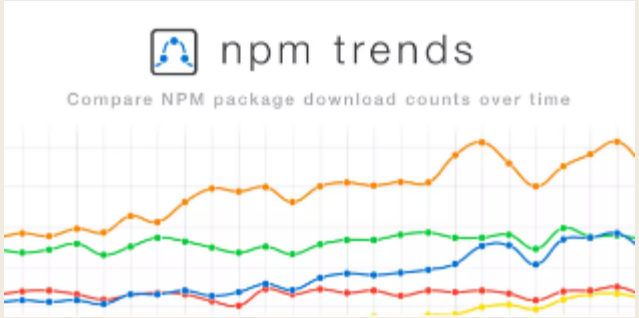
PLAYERS!

DO WE NEED STYLED COMPONENTS?

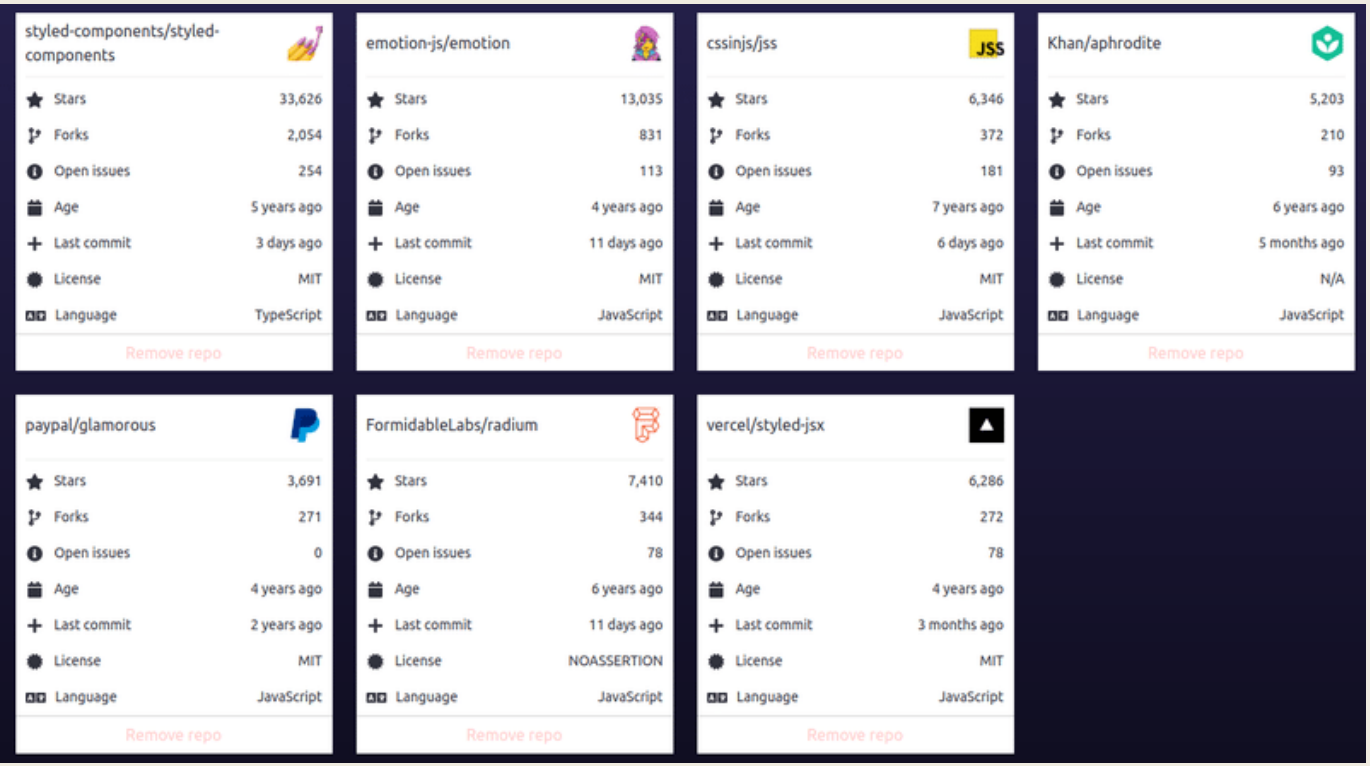
WHY? ALTERNATIVES?

COMPETITION

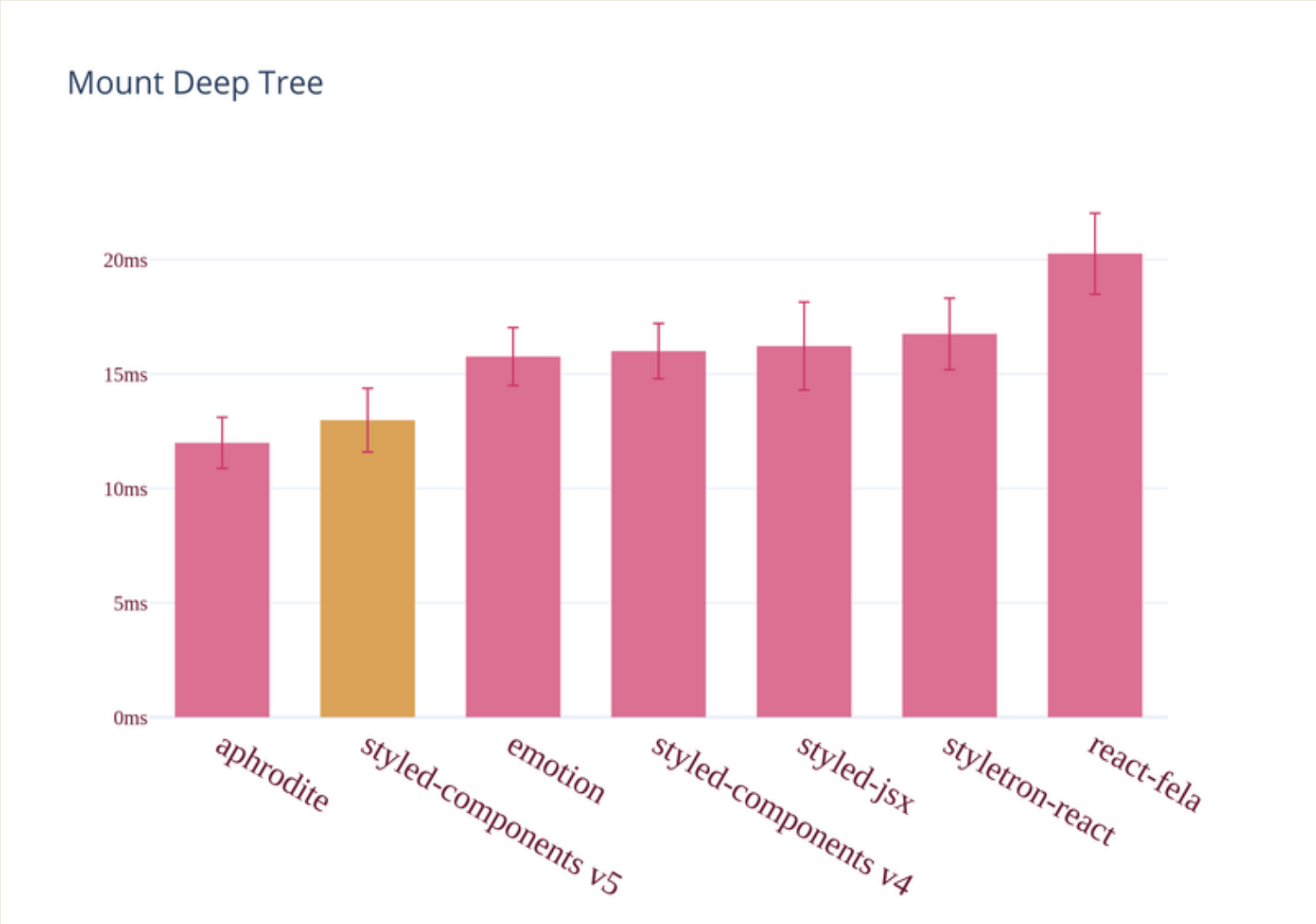




COMPETITION



COMPETITION



ADVANTAGES

- Freedom to build custom components with CSS
- Scoped styles
- Automatic critical CSS
- No-class policy
- Server-side rendering
- Native mobile support
- Sass and polished support
- Automatic vendor prefixing
- Painless maintenance

DISADVANTAGES

- maybe unusual way of styling
- maybe Learning Curve

Tricks

Tricks

PASSING PROPS

LETS MAKE THESE BUTTONS ...

BUTTON

BUTTON

Tricks

PASSING PROPS

```
import styled from 'styled-component';

const Button = styled.button`
  background: ${(props) => (props.bg)};
  color: ${(props) => (props.color )};

  font-size: 1em;
  margin: 1em;
  padding: 0.25em 1em;
  border: 2px solid palevioletred;
  border-radius: 3px;
`;

export default Button;
```

RIGHT?

Tricks

PASSING PROPS

```
import styled from 'styled-component';

const Button = styled.button`
  background: ${({ bg }) => bg};
  color: ${({ color }) => color };

  font-size: 1em;
  margin: 1em;
  padding: 0.25em 1em;
  border: 2px solid palevioletred;
  border-radius: 3px;
`;

export default Button;
```

IS THIS THE BEST ONE?

Tricks

PASSING PROPS

```
import styled from 'styled-component';
import prop from './prop';

const Button = styled.button`
  background: ${prop('bg')};
  color: ${prop('color')};

  font-size: 1em;
  margin: 1em;
  padding: 0.25em 1em;
  border: 2px solid palevioletred;
  border-radius: 3px;
`

export default Button;
```

Tricks

PROP FUNCTION

```
type PredictedProp = {
  [name: string]: any;
};

type PredictedFn = (props: PredictedProp) => number | string | undefined;


export const prop = (name: string, defaultValue?: any): any => {
   return (props: PredictedProp) => props[name] || defaultValue;
};
```

Tricks

USING THEME

```
import styled from 'styled-component';

const Button = styled.button`
  background: ${({ theme }) => (theme.mainBg)};
 color: ${({ theme }) => (theme.mainColor )};


  font-size: 1em;
  margin: 1em;
  padding: 0.25em 1em;
  border: 2px solid palevioletred;
  border-radius: 3px;
`;

export default Button;
```

USING THEME

Tricks

USING THEME

```
import styled from 'styled-component';
import theme from './theme';

const Button = styled.button`
  background: ${theme('mainBg')};
  color: ${theme('mainColor')};

  font-size: 1em;
  margin: 1em;
  padding: 0.25em 1em;
  border: 2px solid palevioletred;
  border-radius: 3px;
`;

export default Button;
```

Tricks

THEME FUNCTION

```
type PredictedProp = {
  [name: string]: any;
};

export const theme = (name: ThemeKey) => (props: PredictedProp) =>
 props.theme[name];
```

Tricks

HOW ABOUT CONDITIONS

```
import styled from 'styled-component';

const Button = styled.button`
  background: ${({ useMainBg, passedBg, theme }) => useMainBg ? passedBg : theme.mainBg};
  color: ${({ useMainColor, passedColor, theme }) => useMaincolor ? passedColor : theme.mainColor};

  font-size: 1em;
  margin: 1em;
  padding: 0.25em 1em;
  border: 2px solid palevioletred;
  border-radius: 3px;
`;

export default Button;
```

TOO MUCH CODE!

Tricks

USING THEME

```
import styled from 'styled-component';
import { IFPTT, theme, prop}  from './helpers';

const Button = styled.button`
  background: ${IFPTT('useMainBg',  prop('passedBg'),  theme('mainBg'))};
  color:  ${IFPTT('useMainColor', prop('passedColor'), theme('mainColor'))};

  /* other styles */
`;

export default Button;
```

Tricks

THEME FUNCTION

```
export const IFPTT = (
  property: string | string[],
  thisStyle: ConditionalStyle,
  thatStyle?: ConditionalStyle
) => {
  return (props: PredictedProp) => {
    let propNames: string[];
    if (typeof property === 'string') {
      propNames = [property];
    } else {
      propNames = property;
    }

    const hasCondition = propNames.every((propName: string) => prop(propName, false)(props));
    if (hasCondition) {
      if (typeof thisStyle !== 'function') return thisStyle;
      return thisStyle(props);
    } else if (thatStyle) {
      if (typeof thatStyle !== 'function') return thatStyle;
      return thatStyle(props);
    }
  };
};
```

HOW STYLED COMPONENTS WORKS?

IF WE FIND IT OUT, WE COULD MAKE MORE BETTER HELPRES

Template literal

```
let output = myTag`That ${ person } is a ${ age }.`;
```

Parameters

```
let person = 'Mike';
let age = 28;

function myTag(strings, personExp, ageExp) {
  let str0 = strings[0]; // "That "
  let str1 = strings[1]; // " is a "
  let str2 = strings[2]; // "."

  let ageStr;
  if (ageExp > 99){
    ageStr = 'centenarian';
  } else {
    ageStr = 'youngster';
  }

  // We can even return a string built using a template literal
  return ${str0}${personExp}${str1}${ageStr}${str2};`
}

let output = myTagThat ${ person } is a ${ age }.;`

console.log(output);
// That Mike is a youngster.
```

Responsiveness

Deside between devices
more smarter

```
import inMobile from './inMobile';

export const ActionsWrapper = styled.div`
  justify-content: space-between;
  display: inline-flex;

  ${inMobile`
  flex: 0 0 7% !important;
  display: flex;
  justify-content: flex-end;

  button {
    margin: 0 !important;
  }
  }
;`
```

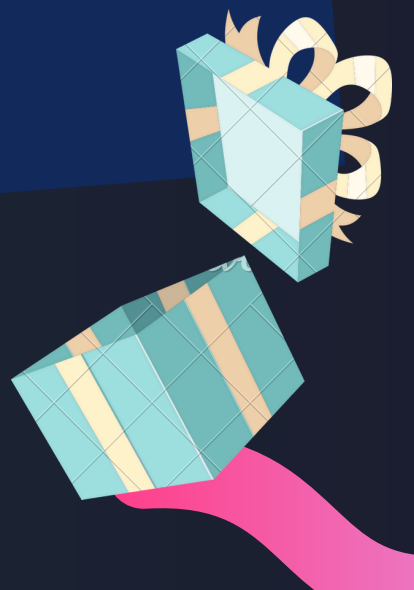
INMOBILE FUNCTION

```
export const inMobile = (styles: any, ...interpolations: any[]) => {
  const deviceType = getMemoizedDeviceType();

  if (deviceType === 'mobile' || deviceType === 'tablet') {
    return renderStyle(styles, interpolations);
  } else {
    return breakpoint('mobile', styles, interpolations);
  }
};
```

RENDER STYLES FUNCTION

```
export const renderStyle = (styles: any, interpolations: any[]) => css`
  ${css(styles, ...interpolations)}
;`
```



RENDER STYLES FUNCTION

```
export const breakpoint = (
  name: keyof typeof breakpoints,
  styles: any,
  ...interpolations: StyledComponentsInterpolation[]
) => {
  const minWidth = breakpoints[name]?.minWidth || 0;
  const maxWidth = breakpoints[name]?.maxWidth || 0;
  let mediaCondition = '';
  if (minWidth) {
    mediaCondition += (min-width: ${minWidth}px);`
  }
  if (maxWidth) {
    mediaCondition += mediaCondition ? ' and ' : '';
    mediaCondition += (max-width: ${maxWidth}px);`
  }

  return css`
    @media ${mediaCondition} {
      ${css(styles, ...interpolations)}
    }
  `;
};
```

Responsiveness

inMobileOnly

inTabletOnly

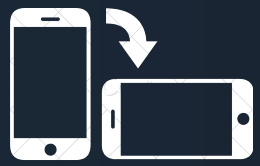
inDesktopOnly

other ideas

space

color

inLandscape



Other features

WITHCOMPONENT

```
const Button = styled.button`
  background-color: blue;
`;

const Link = Button.withComponent('a');
```

REVERSE SELECTOR PATTERN

```
const Link = styled.a`
  display: flex;
  align-items: center;
`;

const Icon = styled.svg`
  width: 48px;
  height: 48px;

  ${Link}:hover & {
    fill: rebeccapurple;
  }
`;
```

DON'T MISS THE BABEL PLUGIN

```
npm install --save-dev babel-plugin-styled-components
```

```
{
  "plugins": ["babel-plugin-styled-components"]
}
```

DEAR ALL

Thank You!

This was SayJeyHi
https://sayjeyhi.com