

DON'T
**STAY
HYDRATED**

JULIAN BURR
@ REACT CONNECTION '24



**GOOD OL'
HTML**

HOW TO USE `<MARQUEE>`
ABSOLUTELY EVERYWHERE

HTML FOR BEGINNERS



HOW TO USE <MARQUEE>
ABSOLUTELY EVERYWHERE

HTML FOR BEGINNERS

NETSCAPE



HOW TO USE <MARQUEE>
ABSOLUTELY EVERYWHERE

HTML FOR BEGINNERS





AJAX & JQUERY

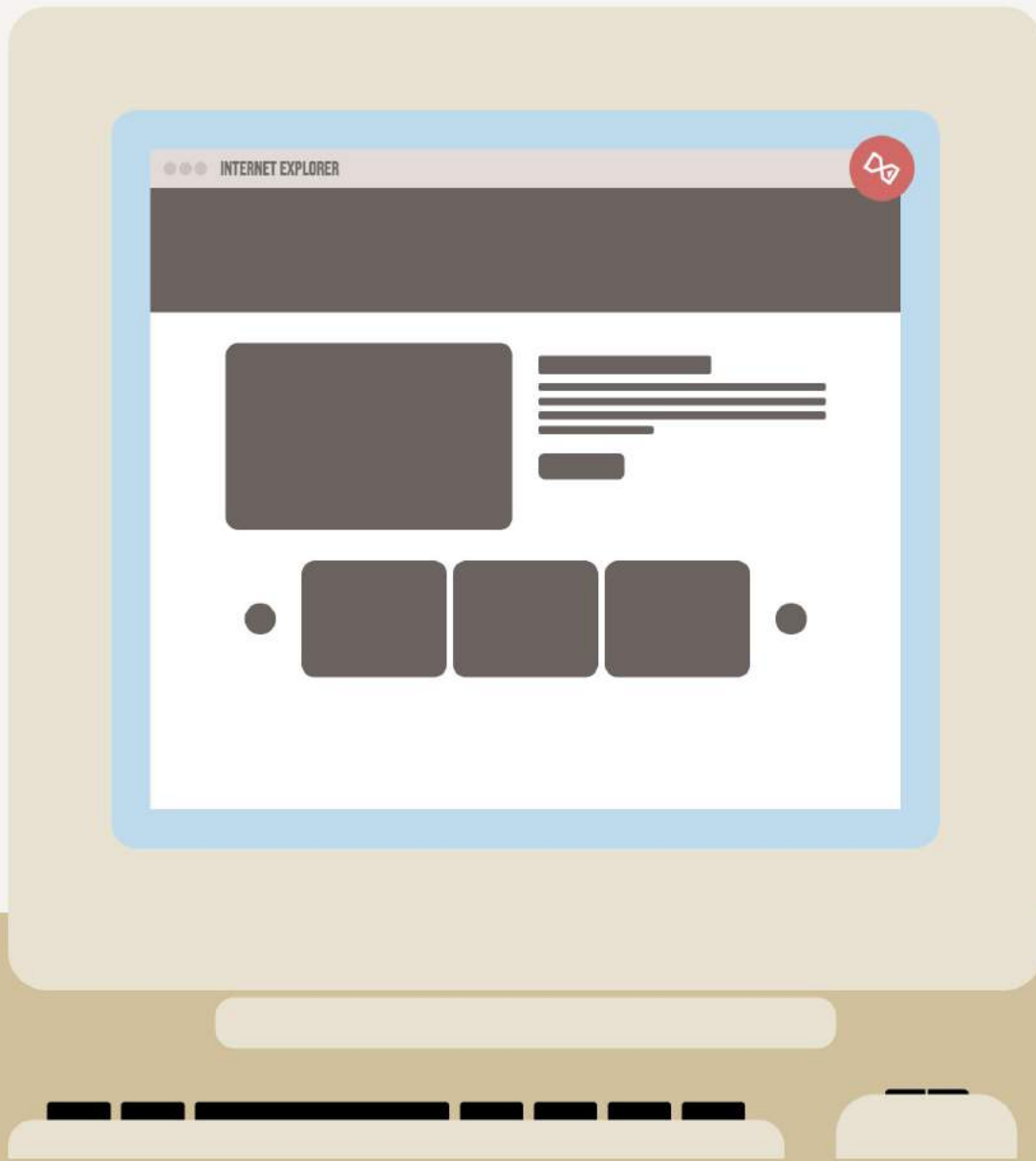
WORDPRESS & MAGENTO

THE MAGIC OF JQUERY



WORDPRESS & MAGENTO

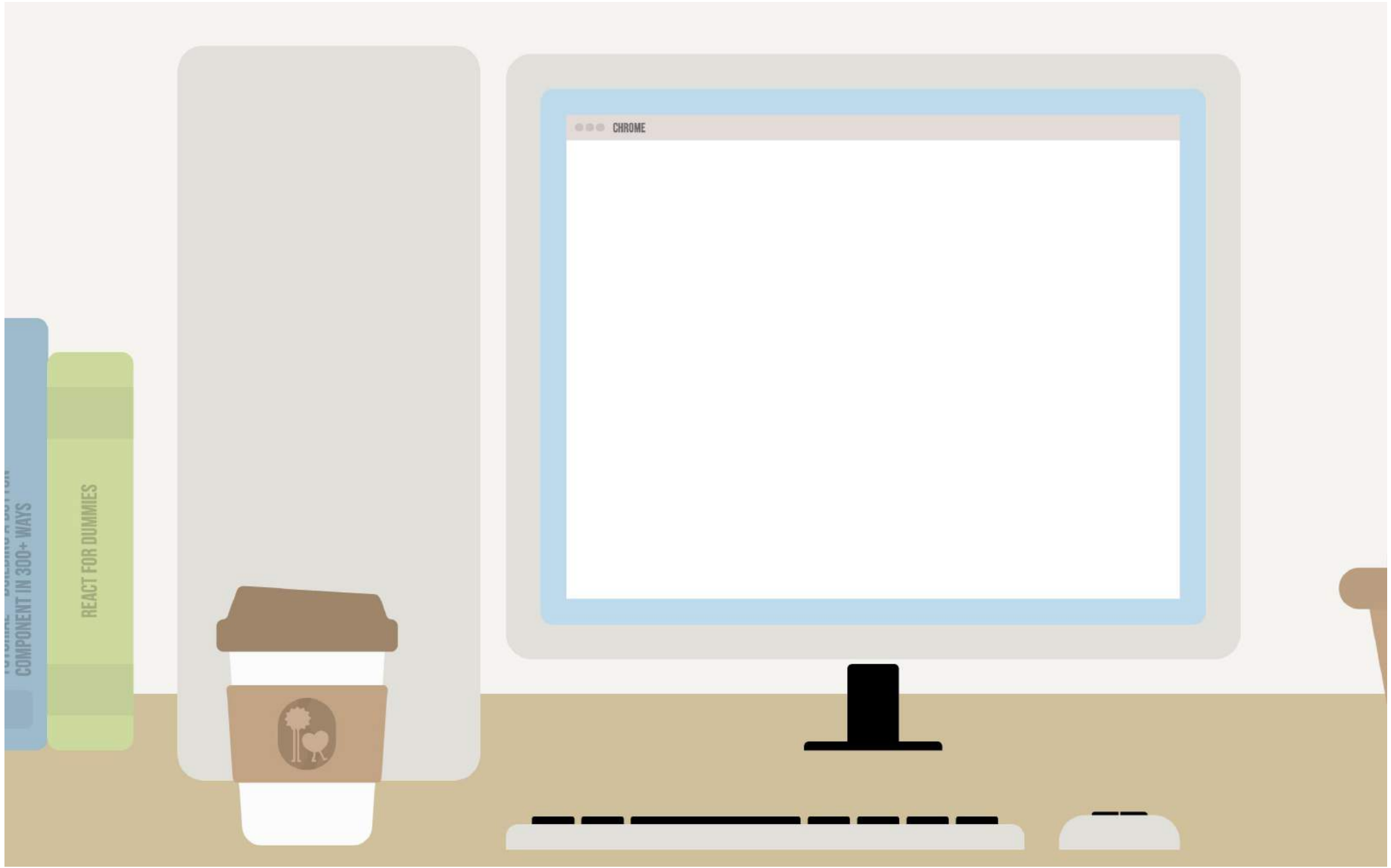
THE MAGIC OF JQUERY





An illustration of a desk setup. On the left, there are two books: a blue one titled 'COMPONENT IN 300+ WAYS' and a green one titled 'REACT FOR DUMMIES'. Next to them is a white coffee cup with a brown sleeve and lid, featuring a logo of a tree and a heart. In the center is a computer monitor with a light blue screen displaying the text 'THE ERA OF SINGLE PAGE APPS'. Below the monitor is a black keyboard and a white mouse. The desk surface is a light brown color.

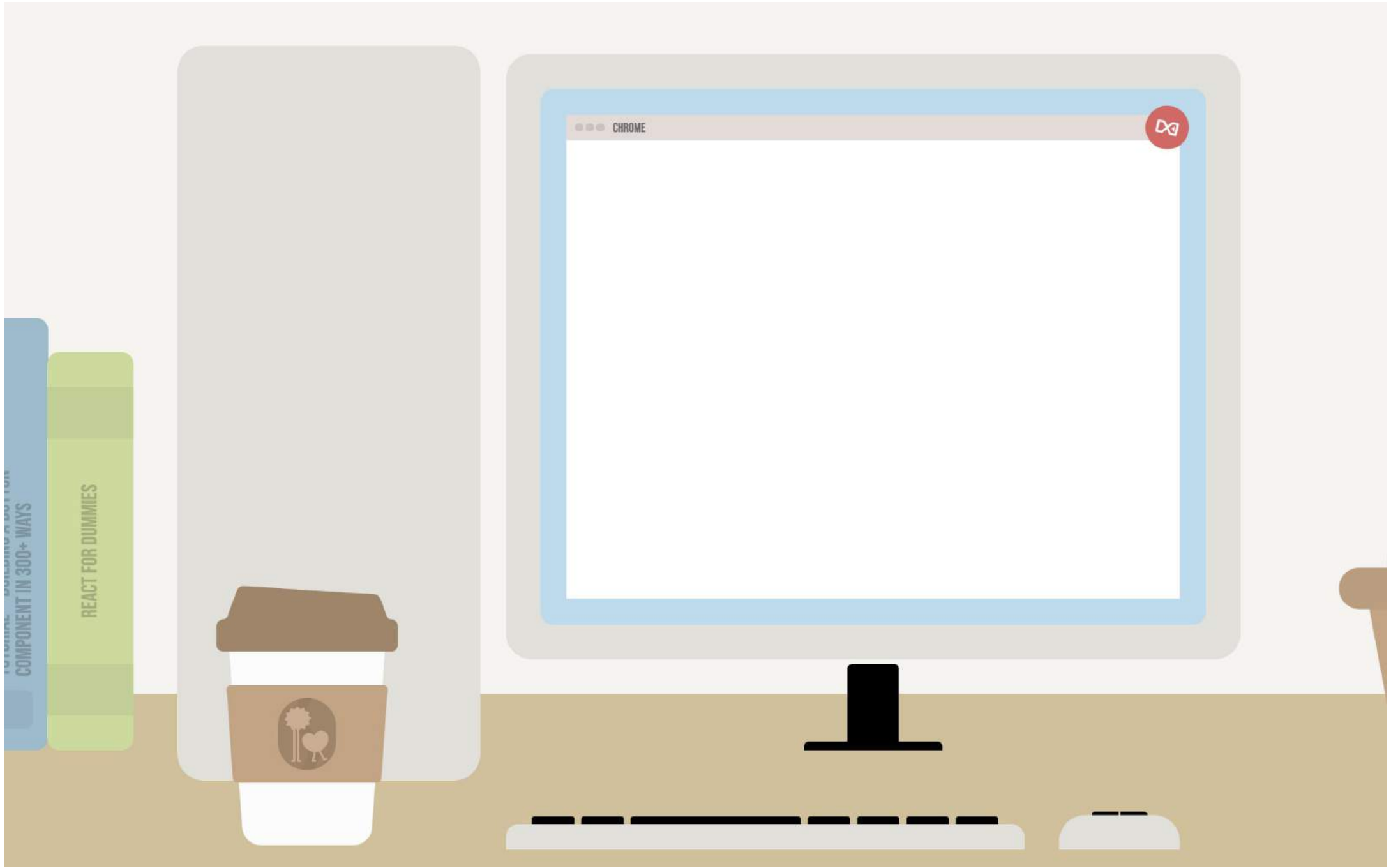
THE ERA OF SINGLE PAGE APPS

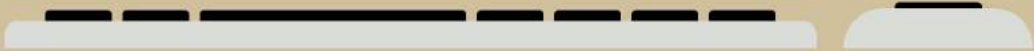


COMPONENT IN 300+ WAYS

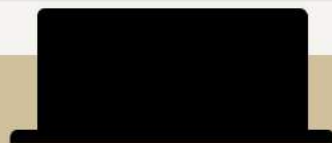
REACT FOR DUMMIES

CHROME





BACK TO THE SERVER









CHROME

← → 1 of 3 unhandled errors

Unhandled Runtime Error

Error: Hydration failed because the initial UI does not match what was rendered on the server.

Call Stack

```
throwOnHydrationMismatch  
  node_modules/react-dom/cjs/react-dom.development.js (12507:0)  
  
tryToClaimHydratableInstance  
  node_modules/react-dom/cjs/react-dom.development.js (12520:0)
```



SERVER COMPONENTS

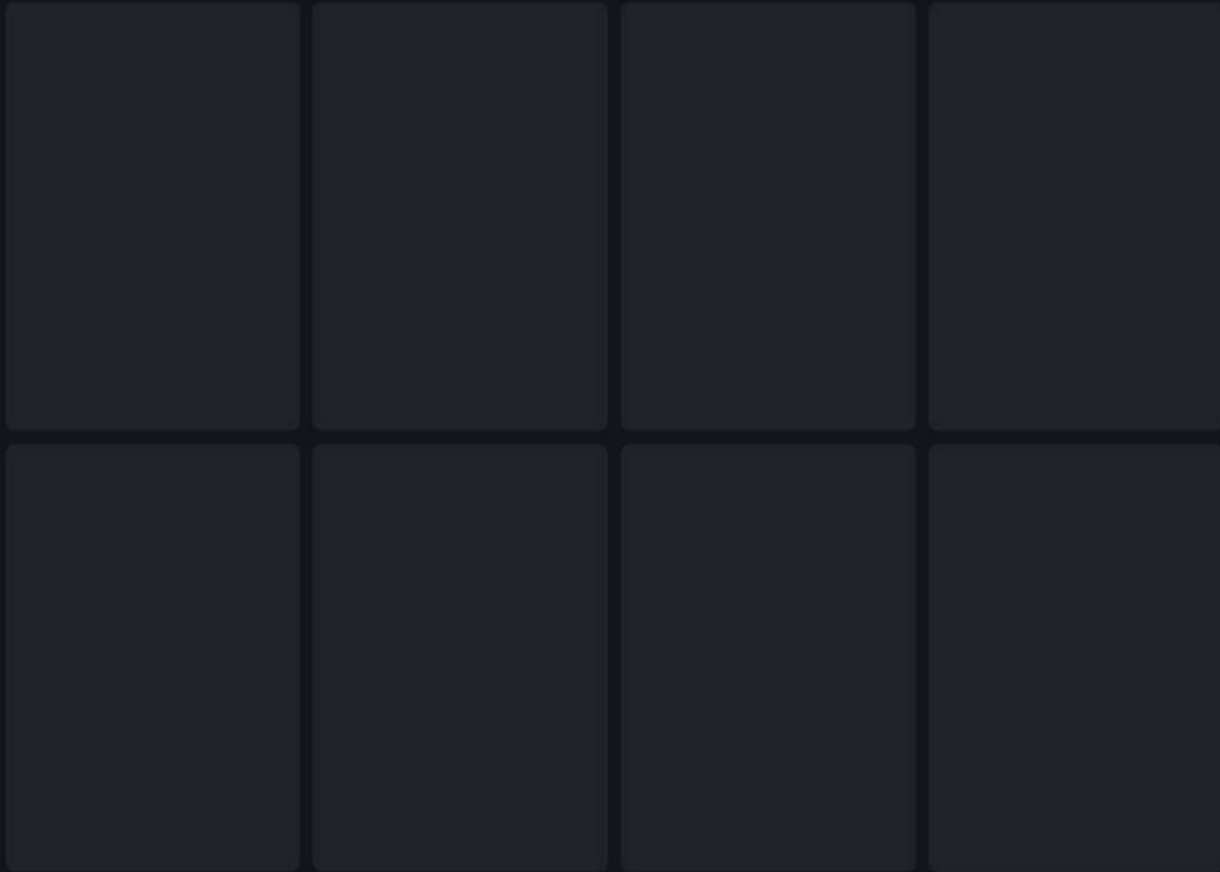
NOTFLIX

🔍 Search movies...



NOTFLIX

🔍 Talent



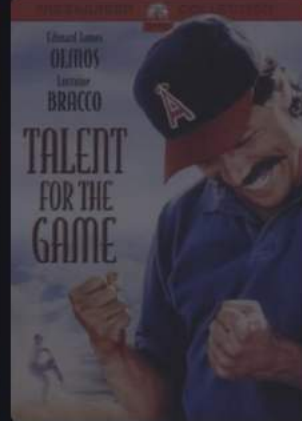
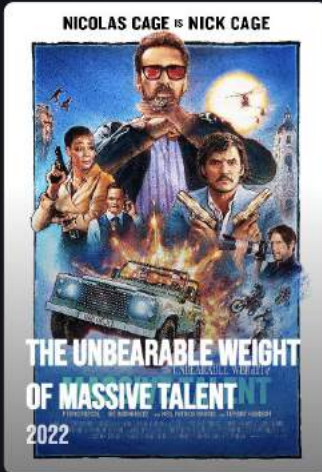
NOTFLIX

🔍 Talent



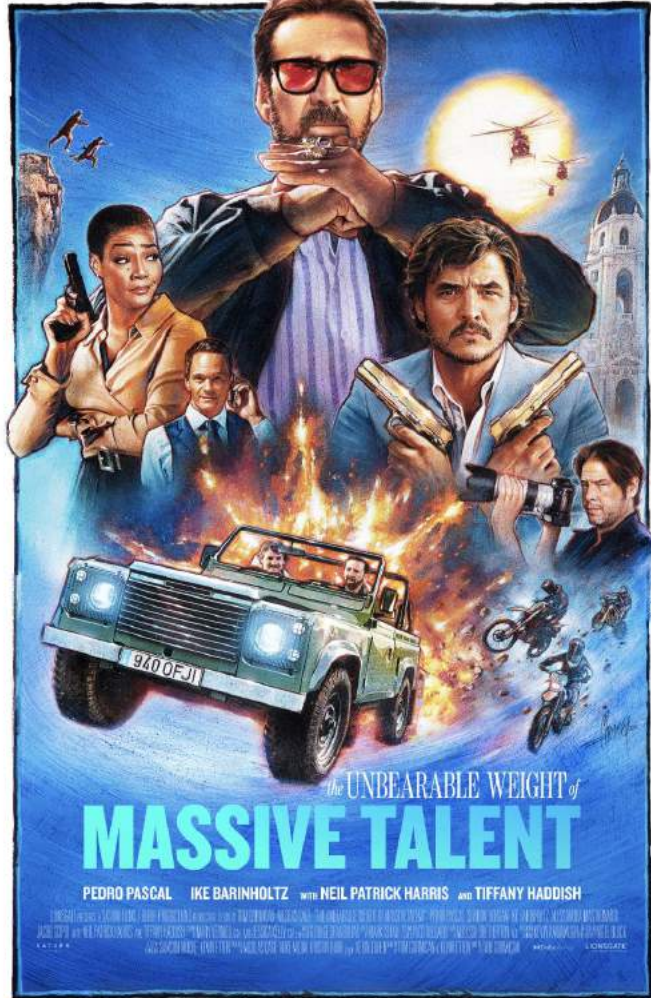
NOTFLIX

🔍 Talent





NICOLAS CAGE IS NICK CAGE



THE UNBEARABLE WEIGHT OF MASSIVE TALENT

2022, 1H 47M
ACTION, COMEDY, CRIME

Moviestar Nick Cage is channeling his iconic characters as he's caught between a superfan and a CIA agent.

DIRECTOR

Tom Gormican

CAST

Nicolas Cage, Pedro Pascal, Tiffany Haddish

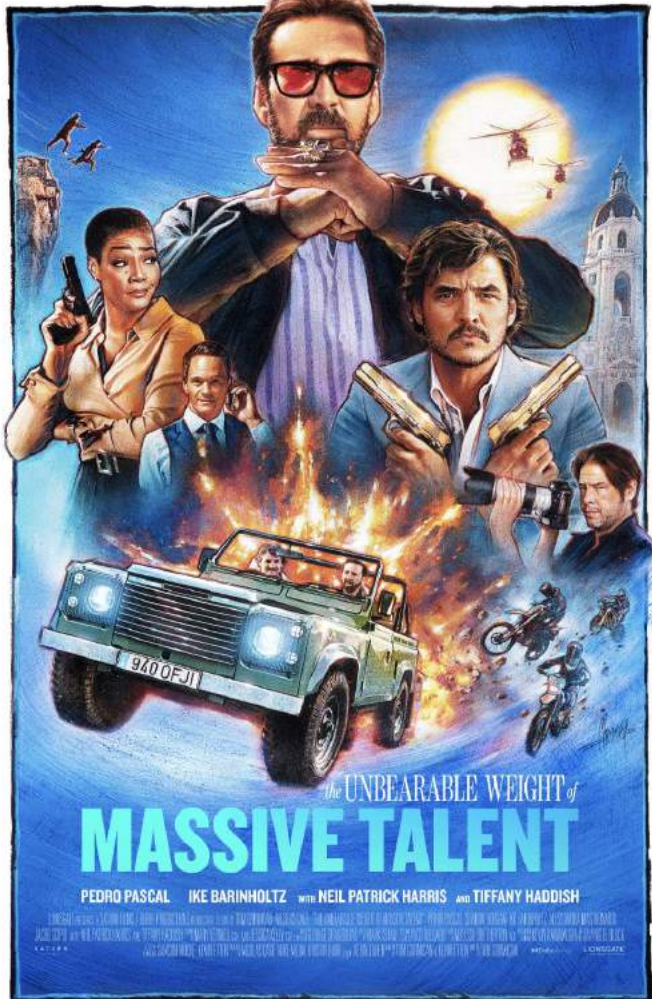
REVIEW: I DID NOT EXPECT THAT!!!

It's fair to say I haven't always been the biggest fan of Nicholas Cage, but it's not the talent, it's often the film choices, similar to Michael Cain, he really has made some shockers. What this film does however is poke a little fun at his past, it's tongue in cheek, but it's a cracking movie.

It's funny, it's energetic, it zips along, there isn't a single dull moment here, you don't need to be a fan or a critic, this movie is just really worth your time.

Loaded with irony, it's plain to see that Cage has a sense of humour, this really is a winner for him.

NICOLAS CAGE IS NICK CAGE



THE UNBEARABLE WEIGHT OF MASSIVE TALENT

2022, 1H 47M

ACTION, COMEDY, CRIME

Moviestar Nick Cage is channeling his iconic characters as he's caught between a superfan and a CIA agent.

DIRECTOR

Tom Gormican

CAST

Nicolas Cage, Pedro Pascal, Tiffany Haddish

REVIEW: I DID NOT EXPECT THAT!!!

It's fair to say I haven't always been the biggest fan of Nicholas Cage, but it's not the talent, it's often the film choices, similar to Michael Cain, he really has made some shockers. What this film does however is poke a little fun at his past, it's tongue in cheek, but it's a cracking movie.

It's funny, it's energetic, it zips along, there isn't a single dull moment here, you don't need to be a fan or a critic, this movie is just really worth your time.

Loaded with irony, it's plain to see that Cage has a sense of humour, this really is a winner for him.

```
1 import markdown from "markdown-it";
2 import moment from "moment";
3
4 function DetailsPage({ data }) {
5   const year = moment(data.short.datePublished).format("YYYY");
6   const description = markdown().render(data.short.review.reviewBody);
7   return (
8     <Main>
9       <Poster data={data} />
10      <Content data={data} year={year} description={description} />
11    </Main>
12  );
13 }
14
15 export const getServerSideProps = async () => {
16   const data = await fetchData({ id: params.movieId });
17   return { props: { data } };
18 }
```

```
1 import markdown from "markdown-it";
2 import moment from "moment";
3
4 function DetailsPage({ data }) {
5   const year = moment(data.short.datePublished).format("YYYY");
6   const description = markdown().render(data.short.review.reviewBody);
7   return (
8     <Main>
9       <Poster data={data} />
10      <Content data={data} year={year} description={description} />
11    </Main>
12  );
13 }
14
15 export const getServerSideProps = async () => {
16   const data = await fetchData({ id: params.movieId });
17   return { props: { data } };
18 }
```

```
1 import markdown from "markdown-it";
2 import moment from "moment";
3
4 function DetailsPage({ data }) {
5   const year = moment(data.short.datePublished).format("YYYY");
6   const description = markdown().render(data.short.review.reviewBody);
7   return (
8     <Main>
9       <Poster data={data} />
10      <Content data={data} year={year} description={description} />
11    </Main>
12  );
13 }
14
15 export const getServerSideProps = async () => {
16   const data = await fetchData({ id: params.movieId });
17   return { props: { data } };
18 }
```

```
1 import storage from "localStorage-slim";
2
3 function LikeButton({ id }) {
4   const defaultActive = storage?.get(id) === "true";
5   const [active, setActive] = useState(defaultActive);
6
7   useEffect(() => storage?.set(id, active ? "true" : "false"), [active]);
8
9   return (
10     <HeartButton
11       active={active}
12       onClick={() => setActive((state) => !state)}
13     />
14   );
15 }
```

```
1 import storage from "localStorage-slim";
2
3 function LikeButton({ id }) {
4   const defaultActive = storage?.get(id) === "true";
5   const [active, setActive] = useState(defaultActive);
6
7   useEffect(() => storage?.set(id, active ? "true" : "false"), [active]);
8
9   return (
10     <HeartButton
11       active={active}
12       onClick={() => setActive((state) => !state)}
13     />
14   );
15 }
```

SSR

NAME	SIZE
the-unbearable-weight-of...	19.2kB
Poppins.woff2	8.1kB
Bungee.woff2	13.5kB
global.css	3.9kB
webpack.js	1.2kB
framework.js	45.7kB
main.js	32.1kB
_app.js	1.3kB
chunk-1.js	19.2kB
chunk-2.js	36.1kB
chunk-3.js	44.6kB
page.js	2.7kB
_buildManifest.js	913B
_ssgManifest.js	422B
index.js	5.1kB

235kB transferred



- 📁 localhost:3000
 - 📁 src/components
 - 📁 node_modules
 - 📁 @remix-run/router
 - 📁 ansi-html-community
 - 📁 core-js-pure
 - 📁 css-loader/dist/runtime
 - 📁 events
 - 📁 feather-icons-react/build
 - 📁 html-entities/lib
 - 📁 linkify-it
 - 📁 localStorage-slim
 - 📁 markdown-it
 - 📁 mdurl
 - 📁 moment
 - 📁 object-assign
 - 📁 prop-types
 - 📁 punycode.js
 - 📁 react
 - 📁 react-dom
 - 📁 react-router
 - 📁 react-router-dom
 - 📁 scheduler


```
1 import markdown from "markdown-it";
2 import moment from "moment";
3
4 function DetailsPage({ data }) {
5   const year = moment(data.short.datePublished).format("YYYY");
6   const description = markdown().render(data.short.review.reviewBody);
7   return (
8     <Main>
9       <Poster data={data} />
10      <Content data={data} year={year} description={description} />
11    </Main>
12  );
13 }
14
15 export const getServerSideProps = async () => {
16   const data = await fetchData({ id: params.movieId });
17   return { props: { data } };
18 }
```

```
1 import markdown from "markdown-it";
2 import moment from "moment";
3
4 async function DetailsPage({ id }) {
5   const data = await fetchData({ id });
6
7   const year = moment(data.short.datePublished).format("YYYY");
8   const description = markdown().render(data.short.review.reviewBody);
9
10  return (
11    <Main>
12      <Poster title={data?.short?.name} year={year} />
13      <Content description={description} />
14    </Main>
15  );
16 }
```

```
1 import markdown from "markdown-it";
2 import moment from "moment";
3
4 async function DetailsPage({ id }) {
5   const data = await fetchData({ id });
6
7   const year = moment(data.short.datePublished).format("YYYY");
8   const description = markdown().render(data.short.review.reviewBody);
9
10  return (
11    <Main>
12      <Poster title={data?.short?.name} year={year} />
13      <Content description={description} />
14    </Main>
15  );
16 }
```

```
1 import markdown from "markdown-it";
2 import moment from "moment";
3
4 async function DetailsPage({ id }) {
5   const data = await fetchData({ id });
6
7   const year = moment(data.short.datePublished).format("YYYY");
8   const description = markdown().render(data.short.review.reviewBody);
9
10  return (
11    <Main>
12      <Poster title={data?.short?.name} year={year} />
13      <Content description={description} />
14    </Main>
15  );
16 }
```

```
1
2
3 import storage from "localStorage-slim";
4
5 function LikeButton({ id }) {
6   const defaultActive = storage?.get(id) === "true";
7   const [active, setActive] = useState(defaultActive);
8
9   useEffect(() => storage?.set(id, active ? "true" : "false"), [active]);
10
11   return (
12     <HeartButton
13       active={active}
14       onClick={() => setActive((state) => !state)}
15     />
16   );
17 }
```

```
1 "use client"
2
3 import storage from "localStorage-slim";
4
5 function LikeButton({ id }) {
6   const defaultActive = storage?.get(id) === "true";
7   const [active, setActive] = useState(defaultActive);
8
9   useEffect(() => storage?.set(id, active ? "true" : "false"), [active]);
10
11   return (
12     <HeartButton
13       active={active}
14       onClick={() => setActive((state) => !state)}
15     />
16   );
17 }
```

SSR

NAME	SIZE
the-unbearable-weight-of...	19.2kB
Poppins.woff2	8.1kB
Bungee.woff2	13.5kB
global.css	3.9kB
webpack.js	1.2kB
framework.js	45.7kB
main.js	32.1kB
_app.js	1.3kB
chunk-1.js	19.2kB
chunk-2.js	36.1kB
chunk-3.js	44.6kB
page.js	2.7kB
_buildManifest.js	913B
_ssgManifest.js	422B
index.js	5.1kB

235kB transferred

RSC

NAME	SIZE
the-unbearable-weight-of...	6.7kB
Poppins.woff2	8.1kB
Bungee.woff2	13.5kB
global.css	3.9kB
webpack.js	2.1kB
framework.js	53.8kB
main.js	809B
_app.js	27.2kB
chunk-1.js	7.4kB
page.js	2.3kB
?rsc=1dvtb	487B

127kB transferred

RSC

NAME	SIZE
■ the-unbearable-weight-of...	6.7kB
■ Poppins.woff2	8.1kB
■ Bungee.woff2	13.5kB
■ global.css	3.9kB
■ webpack.js	2.1kB
■ framework.js	53.8kB
■ main.js	809B
■ _app.js	27.2kB
■ chunk-1.js	7.4kB
■ page.js	2.3kB
■ ?rsc=1dvtb	487B

127kB transferred

RSC

NAME	SIZE
the-unbearable-weight-of...	6.7kB
Poppins.woff2	8.1kB
Bungee.woff2	13.5kB
global.css	3.9kB
webpack.js	2.1kB
framework.js	53.8kB
main.js	809B
_app.js	27.2kB
chunk-1.js	7.4kB
page.js	2.3kB
?rsc=1dvtb	487B

127kB transferred



- 📁 localhost:3000
 - 📁 src/components
 - 📁 node_modules
 - 📁 @remix-run/router
 - 📁 ansi-html-community
 - 📁 core-js-pure
 - 📁 css-loader/dist/runtime
 - 📁 events
 - 📁 feather-icons-react/build
 - 📁 html-entities/lib
 - 📁 linkify-it
 - 📁 localStorage-slim
 - 📁 markdown-it
 - 📁 mdurl
 - 📁 moment
 - 📁 object-assign
 - 📁 prop-types
 - 📁 punycode.js
 - 📁 react
 - 📁 react-dom
 - 📁 react-router
 - 📁 react-router-dom
 - 📁 scheduler

📁 localhost:3000

📁 src/components

📁 node_modules

📁 @swc/helper/esm

📁 localstorage-slim

📁 next

RSC

ASTRO & QWIK

```
1 ---
2 import markdown from "markdown-it";
3 import moment from "moment";
4
5 const id = Astro.params.movieId;
6 const movie = await fetchMovie(id);
7
8 const year = moment(data.short.datePublished).format("YYYY");
9 const description = markdown().render(data.short.review.reviewBody);
10 ---
11
12 <Main>
13   <Poster title={movie?.short?.name} year={year} />
14   <Content description={description} />
15 </Main>
16
17 <script>
18   // ...
19 </script>
```

```
1 ---
2 import markdown from "markdown-it";
3 import moment from "moment";
4
5 const id = Astro.params.movieId;
6 const movie = await fetchMovie(id);
7
8 const year = moment(data.short.datePublished).format("YYYY");
9 const description = markdown().render(data.short.review.reviewBody);
10 ---
11
12 <Main>
13   <Poster title={movie?.short?.name} year={year} />
14   <Content description={description} />
15 </Main>
16
17 <script>
18   // ...
19 </script>
```

```
1 ---
2 import markdown from "markdown-it";
3 import moment from "moment";
4
5 const id = Astro.params.movieId;
6 const movie = await fetchMovie(id);
7
8 const year = moment(data.short.datePublished).format("YYYY");
9 const description = markdown().render(data.short.review.reviewBody);
10 ---
11
12 <Main>
13   <Poster title={movie?.short?.name} year={year} />
14   <Content description={description} />
15 </Main>
16
17 <script>
18   // ...
19 </script>
```



```
1 ---
2 import markdown from "markdown-it";
3 import moment from "moment";
4
5 const id = Astro.params.movieId;
6 const movie = await fetchMovie(id);
7
8 const year = moment(data.short.datePublished).format("YYYY");
9 const description = markdown().render(data.short.review.reviewBody);
10 ---
11
12 <Main>
13   <Poster title={movie?.short?.name} year={year} />
14   <Content description={description} />
15 </Main>
16
17 <script>
18   // ...
19 </script>
```

```
1 export default component$(() => {
2   const loc = useLocation();
3
4   const data = useResource$(async ({ track }) => {
5     track(() => loc.params.movieId);
6
7     const data = await fetchMovie(loc.params.movieId as string);
8     return data;
9   });
10
11   return (
12     <Resource
13       value={data}
14       onPending={() => <Skeleton />}
15       onResolved={(movie) => <Main>{/* ... */}</Main>}
16     />
17   );
18 }
```

```
1 export default component$(() => {
2   const loc = useLocation();
3
4   const data = useResource$(async ({ track }) => {
5     track(() => loc.params.movieId);
6
7     const data = await fetchMovie(loc.params.movieId as string);
8     return data;
9   });
10
11   return (
12     <Resource
13       value={data}
14       onPending={() => <Skeleton />}
15       onResolved={(movie) => <Main>{/* ... */}</Main>}
16     />
17   );
18 }
```

```
1 import storage from "localStorage-slim";
2
3 export default component$(({ id }) => {
4   return (
5     <HeartButton
6       defaultValue={storage.get(id)}
7       onClick$={(e) => {
8         storage.set(id, e.target.checked ? "true" : "false");
9       }}
10    />
11  );
12 }
```

```
1 import storage from "localStorage-slim";
2
3 export default component$(({ id }) => {
4   return (
5     <HeartButton
6       defaultValue={storage.get(id)}
7       onClick$={e => {
8         storage.set(id, e.target.checked ? "true" : "false");
9       }}
10    />
11  );
12 }
```

ARE SERVER COMPONENTS THE FUTURE?



**GITHUB.COM/JULIANBURR/
TALK-DONT-STAY-HYDRATED**

JULIANBURR.DE
GITHUB.COM/JULIANBURR
TWITTER.COM/JBURR90

@ REACT CONNECTION '24