Reactの型、ちゃんと調べる

TypeScript React

childrenの型って何だっけ...?
とかなってしまうので、Reactのコンポーネントの型を調べてまとめます。

ReactのDOMが返す型(children など) って？

ズバリ `React.ReactNode` 。

ReactNodeは下記のように定義されています。

```
type ReactNode =
  ReactChild
  | ReactFragment
  | ReactPortal
  | boolean
  | null
  | undefined
```

union型も辿っていくと、下記の定義が発見できます。

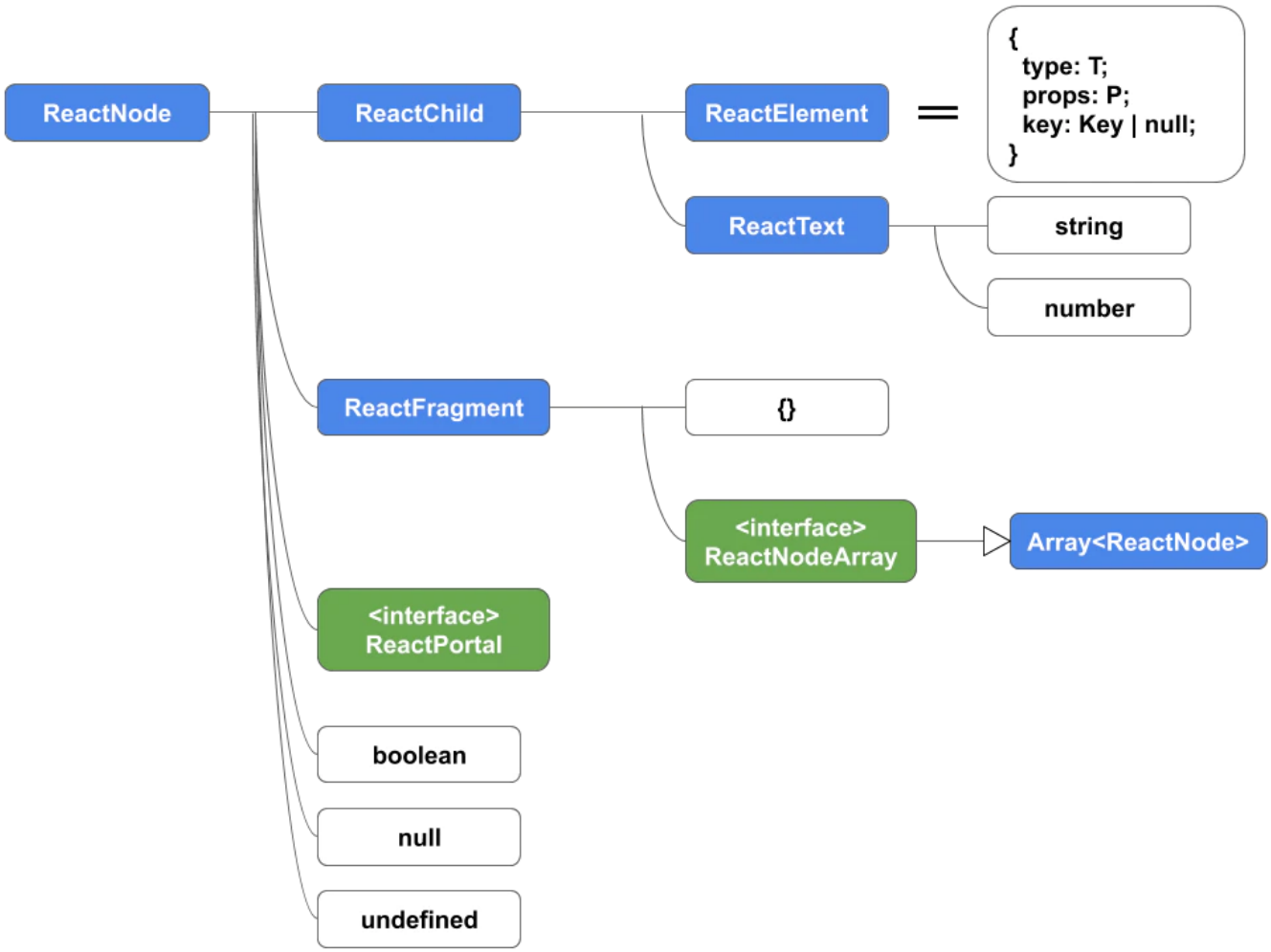
```
type ReactChild = ReactElement | ReactText;
type ReactText = string | number;
interface ReactElement<
  P = any,
  T extends string
  | JSXElementConstructor<any> = string
  | JSXElementConstructor<any>
> {
        type: T;
        props: P;
        key: Key | null;
    }

type ReactFragment = {} | ReactNodeArray;
interface ReactNodeArray extends Array<ReactNode> {}

type JSXElementConstructor<P> =
    | ((props: P) => ReactElement | null)
    | (new (props: P) => Component<P, any>);

interface ReactPortal extends ReactElement {
    key: Key | null;
    children: ReactNode;
}
```

dackdive's blog様に上記を非常にわかりやすくまとめられた図がありましたので、お借りしました👜



それぞれの型定義について以下でより詳しく見てみます。

ReactChild（ReactElement)

```
type ReactChild = ReactElement | ReactText;
type ReactText = string | number;
interface ReactElement<
  P = any,
  T extends string
  | JSXElementConstructor<any> = string
  | JSXElementConstructor<any>
> {
        type: T;
        props: P;
        key: Key | null;
    }
```

ReactElementとはまさに、Reactの原子とも言える型ですね。
propsやkeyを持つ各コンポーネントの型です。
ちなみに `JSX.Element` はReactElementのジェネリックを空で登録したものになります。

```
declare global {
    namespace JSX {
        // tslint:disable-next-line:no-empty-interface
        interface Element extends React.ReactElement<any, any>
...
```

ReactPortal

```
interface ReactPortal extends ReactElement {
    key: Key | null;
    children: ReactNode;
}
```

そもそもPortalという機能を知らなかったのでまとめます。

https://ja.reactjs.org/docs/portals.html

> ポータル (portal) は、親コンポーネントの DOM 階層外にある DOM ノードに対して子コンポーネントをレンダーするための公式の仕組みを提供します。

通常の `render` メソッドで返すと、一番近い親ノードの子として登録されますが、
指定したDOMの子コンポーネントとして挿入できるメソッドのようです。
実際に公式がcodepenでDEMOを載せています。
デモでは外部のモーダルをportalを通して自分の親ではない

DOM要素に生成しています。
このような場合はrefを使うor親でstateを持つしかないと思っていたので斬新な気持ちです。
次回からはこちらでモーダルを生成しようと思いました。

脱線しましたが、 ReactPortal型 とはつまりこの機能のための型です。

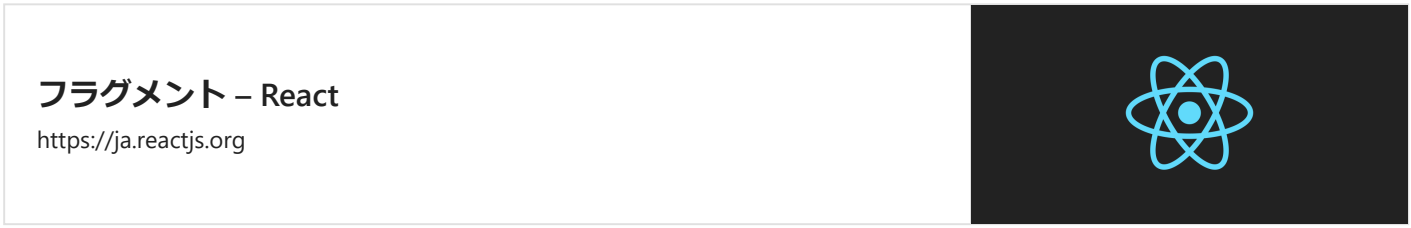
protalのコード

```
render() {
  return ReactDOM.createPortal(
    this.props.children,
    domNode
  );
}
```

createPortalの型定義

```
export function createPortal(
 children: ReactNode,
 container: Element,
 key?: null | string
): ReactPortal;
```

ReactFragment




```
type ReactFragment = {} | ReactNodeArray;
interface ReactNodeArray extends Array<ReactNode> {}
```

これは要素を吐き出したくない時に使うフラグメントの型ですね。

```
return(
  <>
    //ここに各要素
  <>
)
```

{}，もしくはReactNodeの配列を返します。

まとめ

ReactNodeは `boolean` `null` `undefind` なども許容されており、
割となんでも入ってしまう緩い型なので、より厳し目にしたいときはこっちで諸々調整してあげたほうが良さそうです。

参考資料

https://qiita.com/sangotaro/items/3ea63110517a1b66745b#children-%E3%81%AE%E5%9E%8B

https://dackdive.hateblo.jp/entry/2019/08/07/090000