

Rapport – Preact

Inledning:

Preact är ett open-source JavaScript-ramverk som har utvecklats för att erbjuda en snabb och lättviktig lösning vid utveckling av webbapplikationer. Ramverket har en stor mängd likheter med React, men skiljer sig från detta genom en mer effektiv och mindre kod-bas.

<https://preactjs.com/>

Bakgrund:

Preact grundades av Jason Miller i Codepen, då han ville få en djupare förståelse för React, och släpptes officiellt 2015. Sedan dess har Preact blivit ett välrenommerat ramverk inom webbutvecklingssamhället. Det har en stark men relativt liten komunitet bakom sig som ständigt bidrar med förbättringar och tillägg. Preact har även en minimal API, som gör det lätt för utvecklare att förstå och använda, samtidigt som det är mycket kraftfullt och flexibelt ramverk.

<https://preactjs.com/>

<https://jasonformat.com/preact-ama-on-sideway/>

Fördelar:

- + **Enkla API:** Preact har en mycket enkel API, vilket gör det lätt för utvecklare att snabbt sätta sig in i det.
- + **Lättviktig:** Preact har en väldigt liten kod-bas, vilket gör det snabbt att ladda och använda. Detta är särskilt användbart för enheter med begränsad prestanda och för att reducera sidladdningstiden.
- + **God prestanda:** Preact är känt för att ha utmärkt prestanda, vilket gör det till ett utmärkt val för tidskritiska webbapplikationer.
- + **Likheter med React:** Preact har många likheter med React, vilket gör det enkelt för React-utvecklare att hoppa över till Preact.

Nackdelar:

- **Mindre komunitet:** Preact har en mindre komunitet jämfört med React, som kan göra det svårare att hitta svar på problem och få hjälp från andra utvecklare. En mindre komunitet innebär också att ramverkets utvecklingsprocess inte genererar intresse och användning.
- **Begränsad funktionalitet:** Preact har en mindre API jämfört med React, vilket, som tidigare nämnt, gör det lätt att inlära men innebär också att vissa avancerade funktioner saknas.

<https://2022.stateofjs.com/en-US/libraries/front-end-frameworks/>

<https://2018.stateofjs.com/front-end-frameworks/preact/>

Relationen mellan Preact och React:

Preact och React har många likheter i sin syntax, men skiljer sig från varandra i storlek och funktionalitet. React är ett mer omfattande ramverk med en större API, medan Preact är en lättvikigare version av React med en enklare API. Preact är en bra lösning för utvecklare som söker ett snabbt och enkel ramverk för att bygga webbapplikationer, medan React är ett bättre val för mer komplexa projekt där en mer avancerad API och en större komunitet krävs. Jason Miller har i sitt eget yttrade sagt att Preact kommer nog aldrig bli en sammankopplad del av React men hoppas på att deras samarbete utvecklas och att vissa aspekter av Preact implementeras i React.

<https://preactjs.com/>

<https://preactjs.com/guide/v10/differences-to-react/>

<https://jasonformat.com/preact-ama-on-sideway/>

Egna tankar och reflektioner:

React i sig är väldigt nytt för mig och varit ganska svår att förstå vid första blick, vilket var skonsamt då Preact använder sig av samma kod. Inte många skillnader finns mellan de vilket gjorde det lättare att hitta lösningar vid problemlösning eftersom dokumentation kring Preact är begränsad.

```

1  export default function FrameworkRow(props) {
2    normalizeSurvey(props);
3    return (
4      <tr className="frameworkRow" data-cy="chart-row">
5        <td style={{color: props.framework.color}} data-cy="chart-cell">{props.framework.name}</td>
6        <div className="line" style={{borderColor: props.framework.color}} data-cy="row-line"></div>
7        {props.framework.surveys.map((survey) => (
8          <div data-cy="chart-cell">
9            {survey.retention !== "" ? (
10              <div className="circle" style={{borderColor: props.framework.color}} data-cy="chart-circle">
11                {survey.retention + "%"}
12              </div>
13            ) : (
14              <td className="empty-cell" data-cy="chart-cell"></td>
15            )}
16          </div>
17        ))}
18      </tr>
19    );
20    <td style={{color: props.framework.color}} data-cy="chart-cell">{props.framework.name}</td>
21    </tr>
22  );
23  }
24
25  function normalizeSurvey(props) {
26    const missingYears = 7 - props.framework.surveys.length;
27
28    if(missingYears > 0) {
29      for ( let i = 0; i < missingYears; i++) {
30        props.framework.surveys.unshift( {retention: ""});
31      }
32    }
33  }

```

Detta är ett exempel på hur koden ser ut och en person som kan React kommer känna sig bekväm väldigt snabbt med detta ramverk. Detta stycke kod är en så kallad “Component” som dikterar hur applikationen ska presentera datagrafen. Vissa element är fasta och andra framhävs efter en loop av en array och placeras ut. Det finns även ett if-sats i koden som ser lite annorlunda ut för Preacts syntax. En traditionell if-sats kan alltså inte skrivas i Preact syntax och måste istället se ut på detta vis (rad 8-18). Framöver skulle jag nog inte säga att detta kommer vara min personliga preferens av ramverk. Jag kommer nog använda React istället och alternativt använda Preact vid mindre applikationer.

Referenslista:

- Preact's officiella webbplats - <https://preactjs.com/>
- Preact AMA med Jason Miller - <https://jasonformat.com/preact-ama-on-sideway/>
- Stateofjs graf Preact - <https://2022.stateofjs.com/en-US/libraries/front-end-frameworks/> & <https://2018.stateofjs.com/front-end-frameworks/preact/>
- Preact/React skillnader - <https://preactjs.com/guide/v10/differences-to-react/>