**React**

**การสร้างโปรเจค react**npx create-react-app ชื่อโปรเจ็ค  
หลังจากสร้างเสร็จก็ทำการ cd ไปยังโฟเดอร์ที่ตั้งชื่อโปรเจคแล้วพิมพ์ npm start  
หลังจากนั้นเข้าไปที่ localhost:3000 ก็เป็นอันเสร็จ

Graphical user interface

Description automatically generated

**สามารถใช้ตัวแปรเก็บ tag html ได้และมีการเรียกแสดงผลดังนี้**

A screenshot of a computer

Description automatically generated with medium confidence

**Static file** เช่นไฟล์รูปภาพต่างๆเก็บไว้ใน public

**โครงสร้าง**Node\_modules เก็บโมดูลหรือไลบรารี่ที่ทำงานภายในโปรเจคของเรา  
public ไว้เก็บไฟล์ public ต่างๆ เช่น ไฟล์รูปภาพ index.html  
src เก็บ Component หรือ โครงสร้างของ application  
package.json เก็บข้อมูลต่างๆรวมถึง package ที่ใช้ในโปรเจค

Index.html เป็นส่วนที่ใช้แสดงผลโดยทำผ่านทาง index.js ที่ใส่ component ลงไป โดยต้องสร้าง component ขึ้นมา กล่าวคือ index.js เป็นไฟล์ที่เชื่อมระหว่าง index.html และ component ที่จะนำไปแสดงผล

**DOM (Document Object Model)**เมื่อหน้าเว็บโหลดเสร็จเรียบร้อย Web Browser เว็บ browser จะทำการสร้าง dom ของหน้านั้นขึ้นมา โดยมอง html เป็นโครงสร้างต้นไม้  
Diagram

Description automatically generated

**Virtual DOM (Virtual Document Object Model)**เหมือนกับ DOM ปกติเลยแต่จะทำการ copy DOM มาไว้ใน Memory และเมื่อมีการเปลี่ยนแปลงเกิดขึ้นที่ Component จะอัพเดทเฉพาะจุดที่ Component เปลี่ยนแปลง โดยไม่จำเป็นต้องอัพเดท DOM ใหม่ทั้งหมดทำให้ทำงานได้เร็วกว่าปกติ (DOM ปกติจะอัพเดททั้งหน้าเว็บเลย)

**Component**

ส่วนย่อยแต่ละส่วนของหน้าเว็บที่นำมาประกอบกันเหมือนเลโก้แต่ละชิ้นที่เอาไปรวมกันใน app ทีเดียวทำให้สะดวกสำหรับการใช้งาน เพราะ html ส่วนต่างๆจะถูกเก็บแยกเป็น component นั้นเอง  
การสร้าง Component จะสร้างได้ 2 รูปแบบ  
1.Class Component  
2.Function Component

1.Class Component

สร้างคลาสที่ extends มาจาก React.Component และ returnHTML ออกมาแต่ถ้าสร้างเป็น class จะมีความสามารถในการใช้งานมากกว่า Functional

2.Function Component

เป็นรูปแบบการสร้าง function ธรรมดา และ returnHTML ออกมาโดยชื่อ function ที่ใช้สร้าง component ต้องนำหน้าด้วยตัวอักษรตัวใหญ่

A screenshot of a computer

Description automatically generated

Text

Description automatically generatedวิธีการเรียกใช้ component

**React JSX เขียน html ลงใน javascript**

* รูปแบบการเขียน JSX  
  เขียนภายใต้ <div> <section> <Fragment> <> ต้องเขียนภายใต้ tag พวกนี้ถึงจะสามารถใช้งานแบบหลายบรรทัดได้  
  ต้องมีการกำหนด tag เปิด - ปิด ทุกครั้ง
* การใส่ Class Style ที่เป็น Attribute ใน JSX ให้ใช้เป็น className แทน เนื่องจาก class จะไปซ้ำ

**Props**เป็นข้อมูลที่จะถูกส่งไปยัง component แต่ state คือข้อมูลใน component

Graphical user interface, application, Teams

Description automatically generated

วิธีกำหนดแบบที่ 1 ทั่วไป

A screenshot of a computer

Description automatically generated with medium confidence

วิธีเรียกใช้งาน

A screenshot of a computer

Description automatically generated with medium confidence

วิธีกำหนดแบบที่ 2 แบบ object

Graphical user interface, text

Description automatically generated

วิธีกำหนดแบบที่ 3 แบบ array object

Graphical user interface, application

Description automatically generated with medium confidence

วิธีเรียกใช้แบบที่ 3

Graphical user interface, text, application

Description automatically generated

เพิ่มลบดึงข้อมูลใน array ด้วย Push,pop,shift,unshift,join,splice&slice

x.push(ข้อมูล) เพิ่มค่าเข้าไปต่อท้าย array  
pop ถอดค่าสุดท้ายออกจาก array  
shift ถอดค่าตัวหน้าสุดออกจาก array  
unshift เพิ่มค่าตัวหน้าสุดไปใน array  
x.join() แปลงจาก array เป็น string ไม่ได้เปลี่ยนโครงสร้างแค่เปลี่ยน ณ ตอนที่ใช้ โดยแปลงแล้วจะขั้นข้อมูลแต่ละตัวด้วยเครื่องหมาย comma ถ้าอยากจะขั้นด้วยเครื่องหมายอื่นก็สามารถทำได้โดยใส่เครื่องหมายไปในวงเล็บได้เลย  
splice(ต่ำแหน่งที่จะลบ,จำนวนที่จะลบนับรวมตัวที่เลือกด้วย,ข้อมูลที่จะแทนลงในช่องที่ลบ) เป็นการลบและแทนที่  
slice(ต่ำแหน่งเริ่มต้น,ตำแหน่งสุดท้าย-1) ดึงเอาสมาชิกจากตัว array มาใช้

**Loop (for,foreach,forof)**

Foreach ลูปจาก array \*ไม่สามารถเขียน break continue ได้  
array.foreach(element => {  
console.log(element);  
});  
Forof มีการทำงานคล้ายๆกับ foreach และยังสามารถใช้งาน break continue ได้ด้วย

for(const element(ตัวแปรอะไรก็ได้ไม่จำเป็นต้องเป็น element) of x)

       {

           console.log(`อาเรย์นี้มีค่า ${element}`);

       }

การค้นข้อมูลใน array

indexOf(ข้อมูล) => ผลการค้นหาจะได้ตำแหน่ง Index ทึ่ค้นเจอ ถ้าค้นไม่เจอจะได้ -1  
find(ข้อมูล) => ผลการค้นหาจะได้ข้อมูลที่ค้นเจอ ถ้าค้นไม่เจอจะได้ undefined  
โดยในวงเล็บจะใช้เป็น callback function | const search = x.find(element => element===”แดง”)  
findIndex(ข้อมูล) => ผลการค้นหาจะได้ตำแหน่ง Index ทึ่ค้นเจอ ถ้าค้นไม่เจอจะได้ -1  
โดยในวงเล็บจะใช้เป็น callback function | const search = x.findIndex(element => element===”แดง”)

Array map

เป็นการนำเอาค่าสมาชิกภายใน array มา map ค่าและสร้างเป็น array ใหม่ขึ้นมา

const arrayX = [10,20,30,40]  
const arrayY = arrayX.map((e(ค่าใน array),i(indexของarray)) => { return `ค่าใน x ตัวที่ ${i+1} คือ ${e}` })

Array Filter

เป็นการนำเอาค่าสมาชิกภายใน array มากรองค่าและสร้างเป็น array ใหม่ขึ้นมาโดยจะนำมาสร้างใหม่เฉพาะ array ที่ผ่านการคัดกรองเท่านั้นต่างกับ map

const arrayZ = arrayX.filter(e=>{

           if(e>10)

           {

               return e;

           }

       })

สามารถใช้ซ้อน filter ก็ได้

const arrayZ = arrayData.filter(e=>e.salary>10000).filter(e=>e.department === "จอก");

Array reduce

เป็นการนำค่าสมาชิกใน array มาทำการ return ค่ากลับออกไปเป็นค่าเดียว array.reduce((ค่าที่ถูกประมวลผล,element)=>{},ค่าเริ่มต้น)

const totalSalary = arrayData.reduce((salary,e)=>e.salary+salary,0);

console.log(`เงินเดือนรวมทั้งหมด = ${totalSalary}`);

State

คือข้อมูลที่อยู่ใน component ถ้า state เปลี่ยนก็จะทำให้หน้าเว็บเปลี่ยนด้วย