

NPRU

WEBMAIL



ความสัมพันธ์และฟังก์ชัน

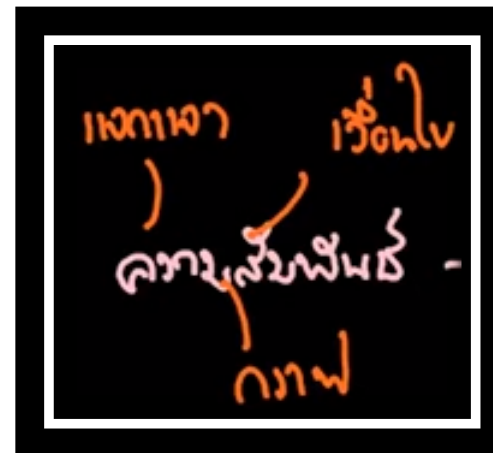
ผู้ช่วยศาสตราจารย์สมเกียรติ ช่อเหมือน (tko@webmail.npru.ac.th)

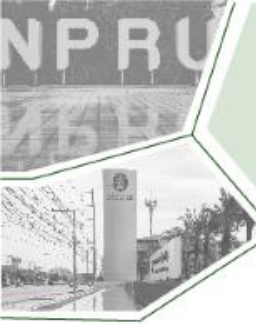
สาขาวิชาวิศวกรรมซอฟต์แวร์ คณะวิทยาศาสตร์และเทคโนโลยี



เนื้อหาการเรียนรู้

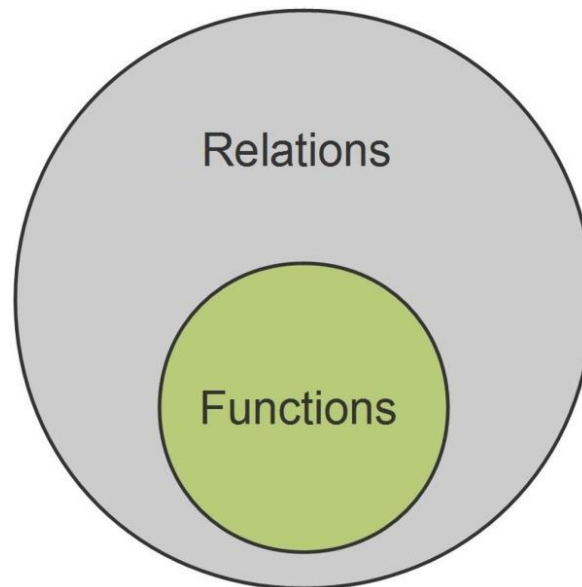
- แบบแผน
- คู่อันดับ
- ผลคูณคาร์ทีเซียน
- ความสัมพันธ์
- โดเมนและเรนจ์
- ตัวผกผันของความสัมพันธ์
- ฟังก์ชัน

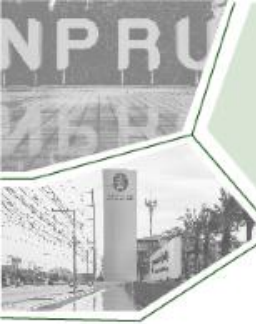




ความสัมพันธ์และฟังก์ชัน (Relation and Function)

- ตัวแทนที่แสดงถึงความสัมพันธ์ของสิ่งที่เกี่ยวข้องกัน ภายใต้กฎเกณฑ์
- ความสัมพันธ์กันของข้อมูลเชิงตัวเลข
- การสร้างฟังก์ชันของคำตอบในรูปแบบสมการ





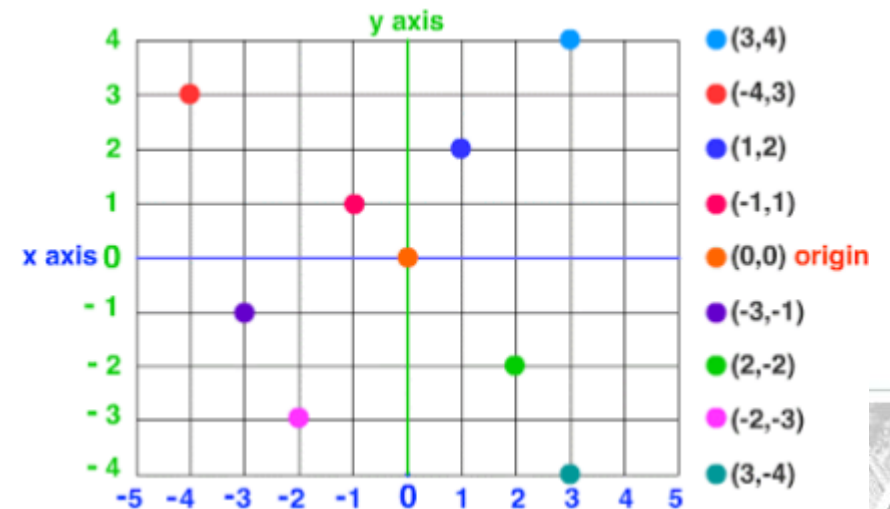
แบบแผน

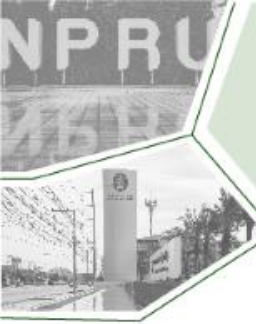
- รูปแบบความสัมพันธ์ (Pattern) ของสิ่งต่าง ๆ มีลักษณะสำคัญ มีเงื่อนไข
- การสังเกต การวิเคราะห์หาเหตุผล บทสรุปที่ยอมรับได้
- การวิเคราะห์แบบแผน ช่วยในการคิดแก้ปัญหา
- กระบวนการวิเคราะห์ด้วยเหตุด้วยผลกับรูปแบบที่แตกต่างกัน
- การจัดลำดับ และการกระทำซ้ำอย่างต่อเนื่องจากการสังเกต
- รูปแบบของจำนวน รูปภาพ รูปทรง เรขาคณิต
- ความสัมพันธ์โดยใช้ตัวแปร และสมบัติของการเท่ากัน มาใช้สร้างสมการ



คู่อันดับ

- คู่อันดับ (Order Pair) คือ การนำสองสิ่งมาเขียนเป็นคู่อันดับ
(3, 2), (-1, 0), (32 , 5), (สมชาย, 5)
- การสลับตำแหน่งของค่าในคู่อันดับ คนละคู่อันดับ เช่น $(3, 2) \neq (2, 3)$
- คู่อันดับจะเท่ากันได้ ก็เมื่อ สมาชิกตัวหน้าและสมาชิกตัวหลังเท่ากัน
- การใช้คู่อันดับในการอ้างอิงหน่วยความจำของอาร์เรย์
 - ตัวชี้และค่า $x[0]=20$
 - คู่อันดับ $(0,20)$



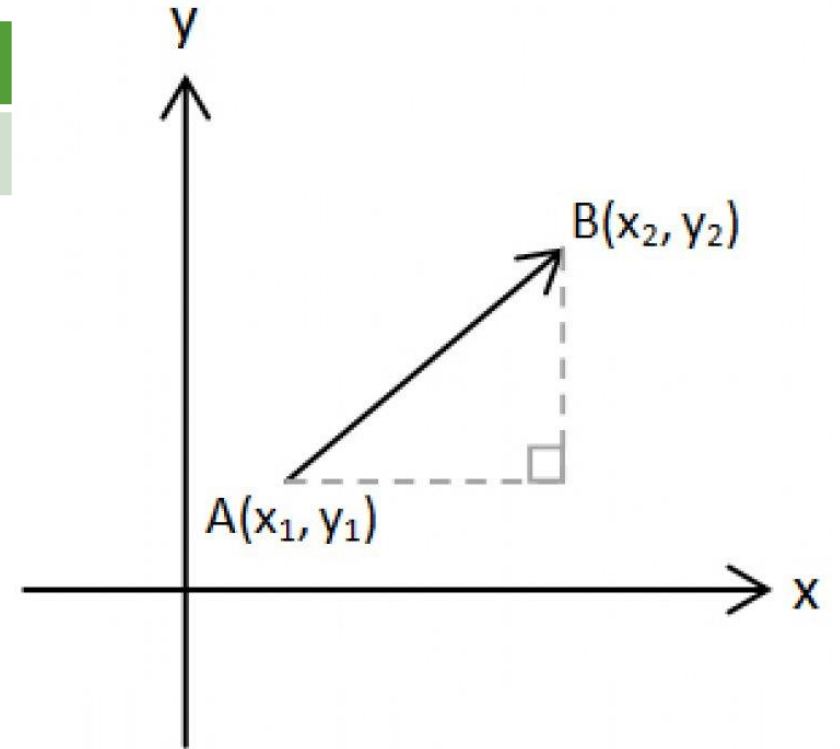


คู่อันดับ

- คู่อันดับยังอาจมองเป็นตาราง เวกเตอร์ 2 มิติ

| | | | | | | |
|-------|---|---|---|---|-----|------|
| ลำดับ | 1 | 2 | 3 | 4 | ... | n |
| จำนวน | 1 | 3 | 5 | 7 | ... | 2n-1 |

- $(1,1), (2,3), (3,4), (4,7), (n,2n-1)$
- การนำคู่อันดับซ้อน $(a, (b, c))$ ของ (a, b, c) ใช้การเวียนเกิด
- ฟังก์ชันเชิงเส้น $y=ax+b$
- กราฟเส้นตรง

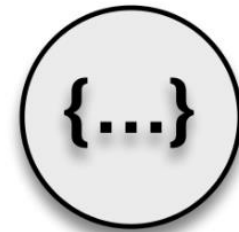


คู่อันดับซ้อน

- (a, b, c) กับ (a, (b, c))
- เซตซ้อน

(a, b, c) และ (a, (b, c))

```
<script>  
let A=['a','b', 'c'];  
let B=[{'a':{'b':'c'}}];  
console.log(A[1]);  
console.log(B[0].a);  
</script>
```



https://www.iconfinder.com/icons/51264/js_icon

<https://tuple.app/pair-programming-guide>

<https://www.slideshare.net/lestermartin/developing-java-streaming-applications-with-apache-storm>

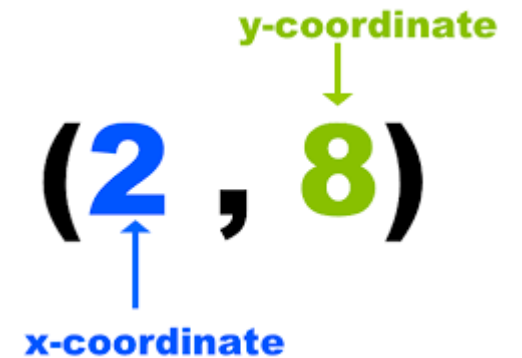
คู่อันดับ $(a, b) = (c, d)$

- **บทนิยาม** คู่อันดับ $(a, b) = (c, d)$ ก็ต่อเมื่อ $a = c$ และ $b = d$ เมื่อ a, b, c, d เป็นจำนวนจริงใด ๆ
 - การหาค่า x และ y ที่ทำให้ $(x + 2, y + 10) = (6, 12)$



$$(x + 2, y + 10) = (6, 12)$$

```
<script>
var x=4;
var y=2;
let a=x+2; //a=c
let b=y+10; //b=d
let c=6;
let d=12;
if(a==c && b==d)
    console.log('(x, y) = ('+x+', '+y+')');
</script>
```



<http://mrsloving.weebly.com/coordinate-plane.html>

https://www.iconfinder.com/icons/51264/js_icon

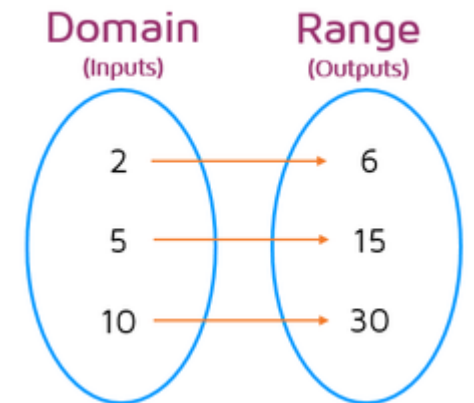
คู่อันดับ $(a, b) = (c, d)$

- การหาค่า x และ y ที่ทำให้ $(2x + y, 24) = (6, 3x - y)$



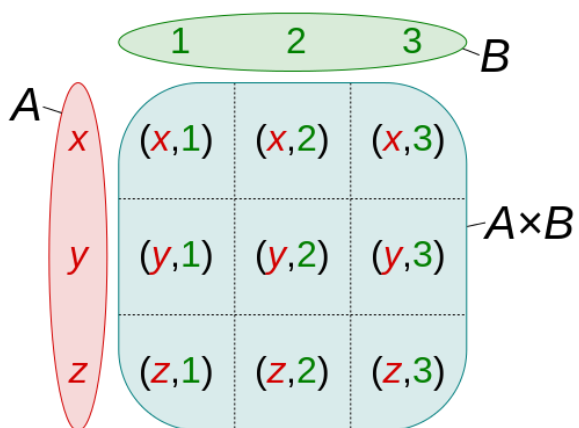
$$(2x + y, 24) = (6, 3x - y)$$

```
<script>
var x=6;
var y=-6;
let a= 2*x + y;
let b=24;
let c=6;
let d= 3*x - y;
if(a==c && b==d)
  console.log('(x,y) = ('+x+', '+y+')');
</script>
```



ผลคูณคาร์ทีเซียน

- ผลคูณคาร์ทีเซียน (Cartesian Product) ของเซต **A** และ **B** เขียนแทนด้วย **A x B**
- เซตแบบเงื่อนไข $A \times B = \{(a, b) \mid a \in A \text{ และ } b \in B\}$
- กำหนดให้ **A = {3, 4}** และ **B = {3, 4, 5}**
- **A x B = {(3, 3), (3, 4), (3, 5), (4, 3), (4, 4), (4, 5)}**



A x B

```
<script>
let A=[3,4];
let B=[3, 4, 5];
let Cartesian =[];
for(let i=0;i<A.length;i++){
  for(let j=0;j<B.length;j++){
    let key_value=('+A[i]+'+'+B[j]+'');
    Cartesian.push(key_value);
  }
}
let result=new Set(Cartesian);
console.log(result);
</script>
```

ความสัมพันธ์

- ความสัมพันธ์ (Relation) คือ เซตของคู่อันดับและเป็นสับเซตของผลคูณคาร์ทีเซียนระหว่างเซต

- r เป็นความสัมพันธ์ แบบกำหนดเงื่อนไข

- $r = A \times B$

r เป็นความสัมพันธ์จาก A ไป B

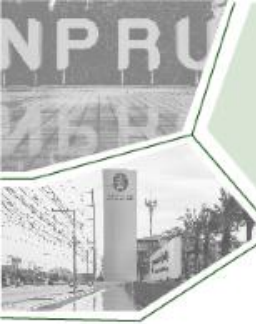
- $r = A \times A$

r เป็นความสัมพันธ์ใน A

- r เป็นเซตของคู่อันดับที่เกี่ยวข้องกันแบบ “น้อยกว่า” จะได้ $r = \{(3, 4), (3, 5), (4, 5)\}$

- r เป็นความสัมพันธ์แบบ “น้อยกว่า” จาก A ไป B เป็นเซตของคู่อันดับที่ได้จากสมาชิกใน $A \times B$





ความสัมพันธ์

- **บทนิยาม** ให้ A และ B เป็นเซต r เป็นความสัมพันธ์จาก A ไป B ก็ต่อเมื่อ r เป็นสับเซตของ $A \times B$

ตัวอย่าง กำหนด $A = \{2, 3\}$, $B = \{4, 6, 9\}$

- r_1 แทนความสัมพันธ์ “สองเท่า” จาก A ไป B
- r_2 แทนความสัมพันธ์ “หารลงตัว” จาก A ไป B
- r_3 แทนความสัมพันธ์ “รากที่สอง” จาก A ไป B

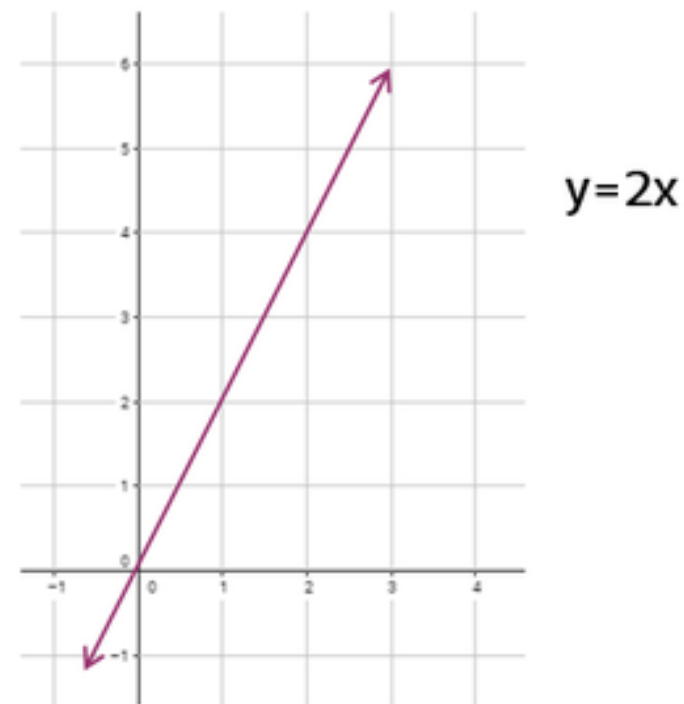
วิธีทำ

$$A \times B = \{(2, 4), (2, 6), (2, 9), (3, 4), (3, 6), (3, 9)\}$$

$$r_1 = \{ \}$$

$$r_2 = \{(2, 4), (2, 6), (3, 6), (3, 9)\}$$

$$r_3 = \{(2, 4), (3, 9)\}$$



เซต r ของความสัมพันธ์

$A \times B = \{(2, 4), (2, 6), (2, 9), (3, 4), (3, 6), (3, 9)\}$

$r_1 = \{ \}$

$r_2 = \{(2, 4), (2, 6), (3, 6), (3, 9)\}$

$r_3 = \{(2, 4), (3, 9)\}$

$A = \{2, 3\}$, $B = \{4, 6, 9\}$ หา r_1, r_2, r_3 (1)

```
<script>
let A=[2, 3];
let B=[4, 6, 9];
let AxB =[];
let r1 =[];
let r2 =[];
let r3 =[];
</script>
```



$A = \{2, 3\}$, $B = \{4, 6, 9\}$ หา r_1, r_2, r_3 (2)

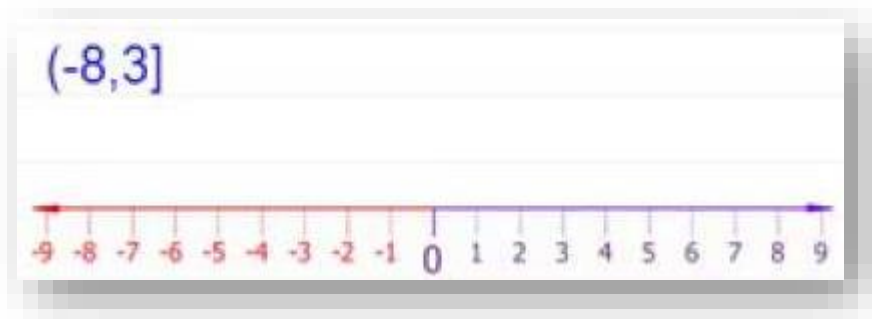
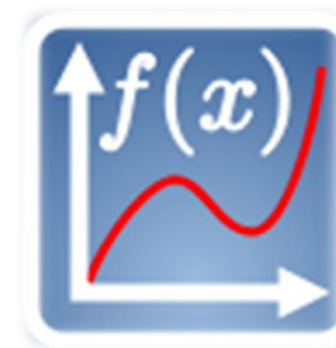
```
<script>
for(let i=0;i<A.length;i++){
  for(let j=0;j<B.length;j++){
    let key_value='('+A[i]+' '+B[j]+' )';
    AxB.push(key_value);
    if(A[i]== 2*B[j]) //r1
      r1.push(key_value);
    if(B[j]%A[i]==0) //r2
      r2.push(key_value);
    if(A[i]== Math.sqrt(B[j])) //r3
      r3.push(key_value);
  }
}
</script>
```

$A = \{2, 3\}$, $B = \{4, 6, 9\}$ หา r_1, r_2, r_3 (3)

```
<script>
let result=new Set(AxB);
console.log(result);
let result_r1=new Set(r1);
console.log(result_r1);
let result_r2=new Set(r2);
console.log(result_r2);
let result_r3=new Set(r3);
console.log(result_r3);
</script>
```

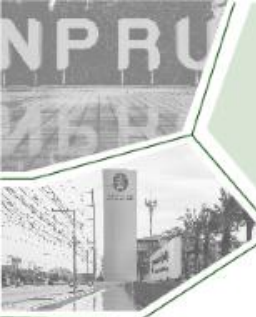
โดเมนและเรนจ์

- โดเมน ของความสัมพันธ์
 - เซตของสมาชิกตัวหน้าของคู่อันดับทั้งหมด **พิกัด x**
 - $D_r = \{(x, y) \mid x \in R\}$
- เรนจ์ ของความสัมพันธ์
 - เซตของสมาชิกตัวหลังของคู่อันดับทั้งหมด **พิกัด y**
 - $R_r = \{(x, y) \mid y \in R\}$
- เงื่อนไขความสัมพันธ์ $f(x) = ax + b$



https://play.google.com/store/apps/details?id=fahim_edu.matematikasd&hl=th

<https://www.youtube.com/watch?v=x1bt-Maf0WI>

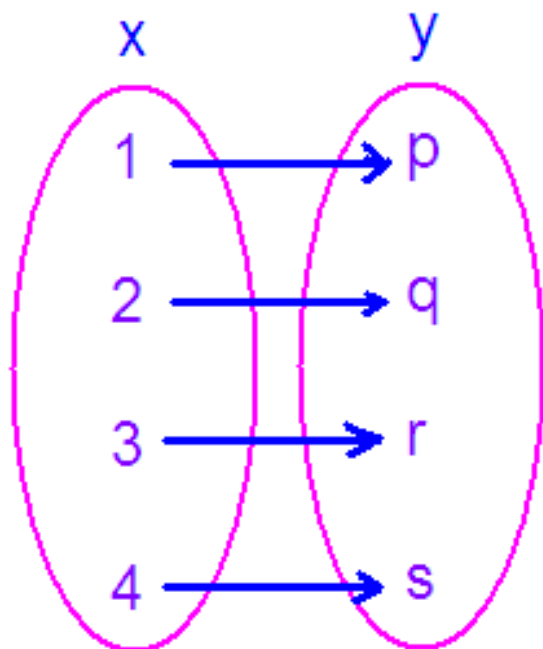


แผนภาพความสัมพันธ์

$r = \{(1,p), (2,q), (3,r), (4,s)\}$

$Dr = \{1, 2, 3, 4\}$

$Rr = \{p, q, r, s\}$



โดเมน และ เรนจ์ของความสัมพันธ์

```
<script>
let Dr=[1,2,3,4];
let Rr=['p','q','r','s'];
let set = [];
for(let i=0;i<=3;i++){
let obj = {};
obj[Dr[i]] = Rr[i]; //จับคู่อันดับ
set.push(obj);
}console.log(set);
</script>
```

https://www.iconfinder.com/icons/51264/js_icon

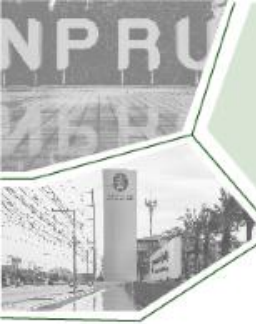


ตัวผกผันของความสัมพันธ์

- ตัวผกผันของความสัมพันธ์ (Inverse of Relation)
- r คือความสัมพันธ์ซึ่งเกิดจากการสลับที่ของสมาชิกตัวหน้าและสมาชิกตัวหลังในแต่ละคู่อันดับ
- r ตัวผกผันของความสัมพันธ์ r เขียนแทนด้วย r^{-1}

$r = \{(1,1), (3,2), (1,3), (4,1), (0,-1)\}$ จะได้ $r^{-1} = \{(1,1), (2,3), (3,1), (1,4), (-1,0)\}$

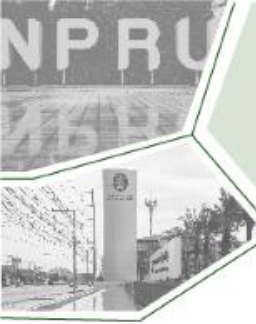




ฟังก์ชัน (Function)

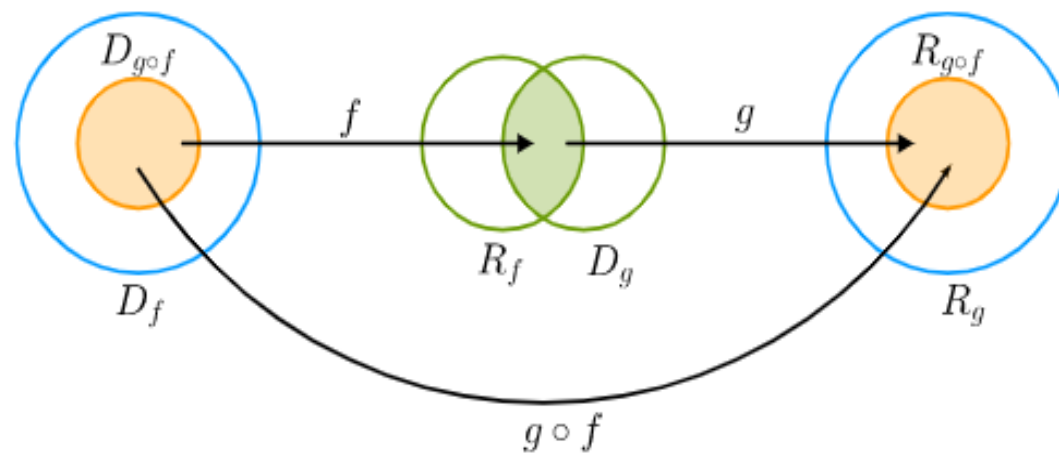
- ฟังก์ชัน คือ ความสัมพันธ์ ซึ่งในสองคู่อันดับใด ๆ ของความสัมพันธ์นั้น
- ความสัมพันธ์ของสองคู่อันดับใด ๆ
- สมาชิกตัวหน้าเท่ากัน สมาชิกตัวหลังต้องเท่ากัน
- การดำเนินการของฟังก์ชัน
- พีชคณิตของฟังก์ชัน นำฟังก์ชันตั้งแต่สองขึ้นไปมาดำเนินการกัน เป็นฟังก์ชันใหม่

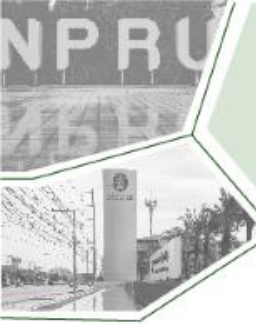




ฟังก์ชันประกอบ

- **บทนิยาม** ให้ f และ g เป็นฟังก์ชันค่าจริง เรานิยามฟังก์ชันผลบวก ฟังก์ชันผลลบ ฟังก์ชัน ผลคูณ ฟังก์ชันผลหาร และฟังก์ชันประกอบ (**composite function**) ระหว่าง f และ g ตามลำดับ





ฟังก์ชันจากเซตหนึ่งไปยังอีกเซตหนึ่ง

- $f : A \rightarrow B$

- ความสัมพันธ์ของเซต

$$A = \{1, 2, 3, 5\}$$

$$B = \{0, 1, 2, 3, 4\}$$

$$f = \{(1, 2), (2, 0), (3, 4), (5, 2)\}$$

- f เป็นฟังก์ชันจาก A ไป B

- $f : A \rightarrow B$

- ฟังก์ชันจาก A ไปทั่วถึง B

- ฟังก์ชันหนึ่งต่อหนึ่งจาก A ไป B

- ฟังก์ชันหนึ่งต่อหนึ่งจาก A ไปทั่วถึง B



ฟังก์ชันเชิงเส้น

- ฟังก์ชันเชิงเส้น (linear function) $y=ax + b$ เมื่อ a, b เป็นจำนวนจริง และ $a \neq 0$



$$f(x)=4x + 7$$

```
<script>
x=[0,1,4,7,9];
fx=[];
for(let i=0;i<=4;i++){
fx[i] = (4* x[i]) + 7; //y=ax+b
console.log(fx[i]);
}
</script>
```

ฟังก์ชันพหุนาม

- ฟังก์ชันพหุนาม (polynomial function) นิพจน์ที่สร้างจากตัวแปรอย่างน้อยหนึ่งตัวและสัมประสิทธิ์ โดยใช้การดำเนินการแค่ การบวก การลบ การคูณ และการยกกำลังโดยที่เลขชี้กำลังเป็นจำนวนเต็มที่ไม่เป็นลบเท่านั้น



$$f(x) = x^2 - 4x + 7$$

```
<script>
x=[0,1,4,7,9];
fx=[];
for(let i=0;i<=4;i++){
fx[i] = Math.pow(x[i], 2) - (4* x[i]) + 7;
console.log(fx[i]);
}
</script>
```

ฟังก์ชันขั้นบันได

- ฟังก์ชันขั้นบันได (step function) ฟังก์ชันของจำนวนจริงซึ่งเกิดจากการรวมกันระหว่างฟังก์ชันคงตัวจากโดเมนที่แบ่งออกเป็นช่วงหลายช่วง

| ลำดับ | เกณฑ์น้ำหนัก (กรัม) | อัตราค่าบริการ (บาท) |
|-------|-----------------------------|----------------------|
| 1 | ไม่เกิน 20 | 2 |
| 2 | เกิน 20 แต่ไม่เกิน 100 | 3 |
| 3 | เกิน 100 แต่ไม่เกิน 250 | 5 |
| 4 | เกิน 250 แต่ไม่เกิน 500 | 9 |
| 5 | เกิน 500 แต่ไม่เกิน 1,000 | 16 |
| 6 | เกิน 1,000 แต่ไม่เกิน 2,000 | 30 |

ฟังก์ชันขั้นบันได

```
<script>
function getPriceW(x) {
  if(x<=2000 && x>1000)
    return 30;
  else if(x>500)
    return 16;
  else if(x>250)
    return 9;
  else if(x>100)
    return 5;
  else if(x>20)
    return 3;
  else return 2;
}
</script>
```



ฟังก์ชันเอกซโพเนนเชียล

- ฟังก์ชันเอกซโพเนนเชียล (exponential function) $f = \{(x, y) \in \mathbb{R} \times \mathbb{R}^+ \mid y = a^x, a > 0, a \neq 1\}$
- ฟังก์ชันเอกซโพเนนเชียลผ่านไลบรารี `Math.exp()`



ฟังก์ชันเอกซโพเนนเชียล

```
<script>
console.log(Math.exp(0));
// expected output: 1

console.log(Math.exp(1));
// expected output: 2.718281828459 (approximately)

console.log(Math.exp(-1));
// expected output: 0.36787944117144233

console.log(Math.exp(2));
// expected output: 7.38905609893065
</script>
```

ฟังก์ชันลอการิทึม

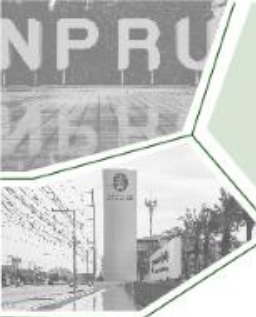
- ฟังก์ชันลอการิทึม (logarithm function) $y = \log_a x$; $a > 0$, $a \neq 1$
- ฟังก์ชันในการหาค่าลอการิทึมใช้งาน `Math.log()` ด้วยการระบุฐาน (a) และตัวค่า (x)



ฟังก์ชันลอการิทึม

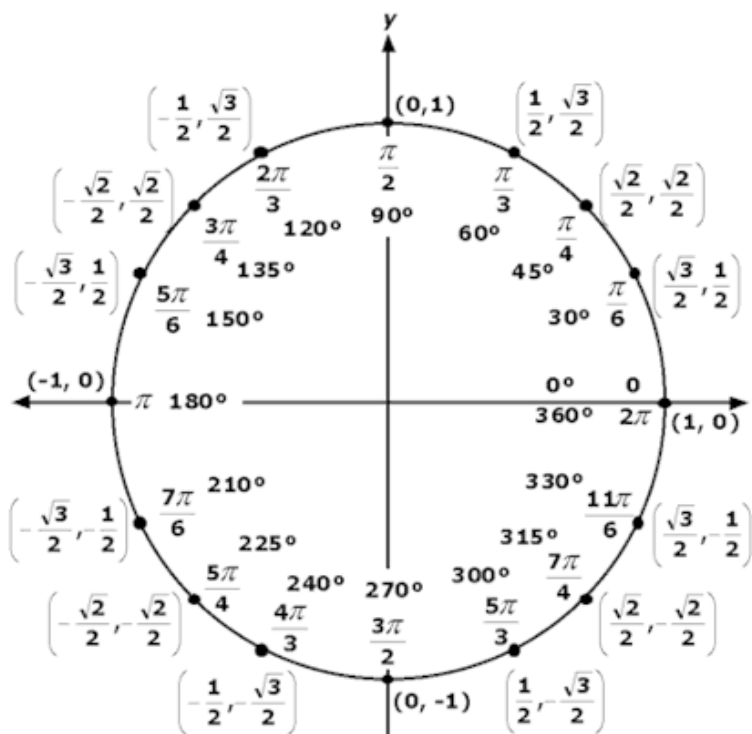
```
<script>
function getBaseLog(x, y) {
  return Math.log(y) / Math.log(x);
}

// 2 x 2 x 2 = 8
console.log(getBaseLog(2, 8));
// expected output: 3
</script>
```

ฟังก์ชันตรีโกณมิติ

- ฟังก์ชันตรีโกณมิติ (trigonometry function) การศึกษารูปสามเหลี่ยมและลักษณะของคาบ อัตราส่วนของด้าน 2 ด้านของรูปสามเหลี่ยมมุมฉาก อัตราส่วนของพิกัดของจุดบนวงกลมหนึ่งหน่วย ผลรวมของมุมทุกมุมจึงเท่ากับ 180° เสมอของค่าเรเดียน



https://www.iconfinder.com/icons/51264/js_icon

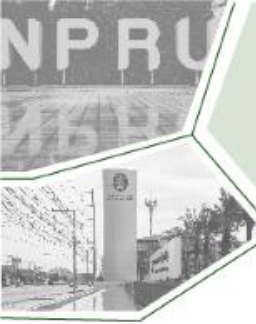
```

ฟังก์ชันตรีโกณมิติ

<script>
function getCircleY(radians, radius) {
  return Math.sin(radians) * radius;
}
console.log(getCircleY(1, 10));
// expected output: 8.414709848078965
console.log(getCircleY(2, 10));
// expected output: 9.092974268256818
console.log(getCircleY(Math.PI, 10));
// expected output: 1.2246467991473533e-15
</script>

```

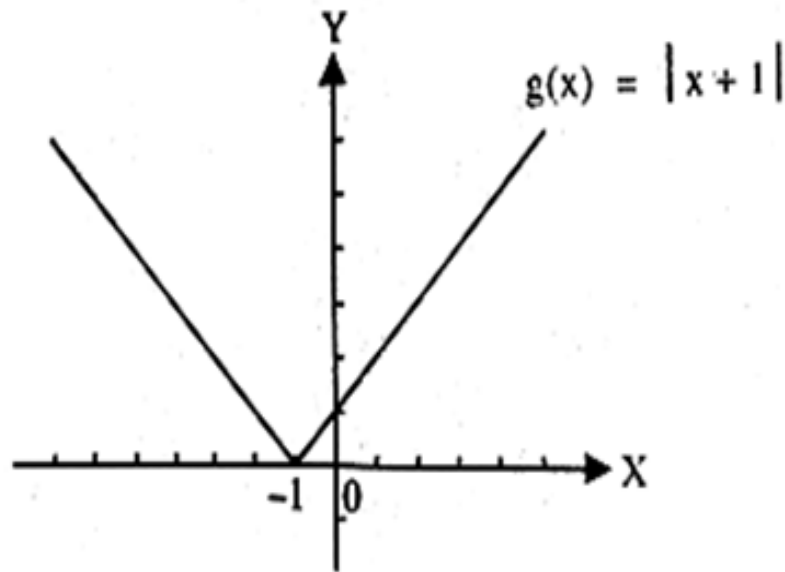




ฟังก์ชันค่าสัมบูรณ์

- ฟังก์ชันค่าสัมบูรณ์ (absolute value function) $y = |x - a| + c$ เมื่อ a และ c เป็นจำนวนจริง ในรูปแบบตารางและกราฟ

| x | g(x) |
|----|------|
| -3 | 2 |
| -2 | 1 |
| -1 | 0 |
| 0 | 1 |
| 1 | 2 |
| 2 | 3 |
| 3 | 4 |



$$y = |x - a| + c$$

```
<script>
x= [-3,-2,-1,0,1,2,3];
gx=[];
for(let i=0;i<=6;i++){
gx[i]=Math.abs(x[i]+1);
console.log(gx[i]);
}
</script>
```



https://www.iconfinder.com/icons/51264/js_icon



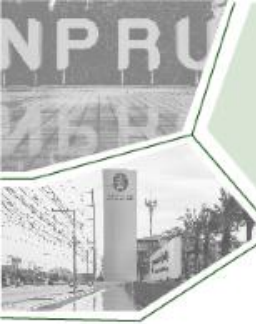
ฟังก์ชันเวียนเกิด

- ฟังก์ชันเวียนเกิด (recursive function) ฟังก์ชันที่เรียกตัวเอง หลักการฟังก์ชันเวียนบังเกิดคือ เขียนโปรแกรมวนซ้ำเพื่อลดปัญหาของโปรแกรมที่ซับซ้อน



ฟังก์ชันเวียนเกิด

```
<script>
function factorial(x) {
  if (x < 0) return;
  if (x === 0) return 1;
  return x * factorial(x - 1);
}
console.log( factorial(5) );
//5*4*3*2*1 5! =120
</script>
```



สรุปท้ายบท

- การวิเคราะห์ความสัมพันธ์ที่มีแบบแผนหรือลักษณะที่มีเงื่อนไข
- อธิบายความสัมพันธ์โดยหาเหตุผล เปรียบเทียบคู่อันดับ หรือเซตสมาชิกตัวหน้ากับสมาชิกตัวหลัง
- ผลคูณคาร์ทีเซียน และฟังก์ชันของคำตอบ ซึ่งถูกนำไปใช้ในการแก้ปัญหา
- ฟังก์ชันประกอบ ฟังก์ชันอินเวอร์ส ฟังก์ชันจากเซต ฟังก์ชันเชิงเส้น ฟังก์ชันพหุนาม ฟังก์ชัน
ขั้นบันได ฟังก์ชันเอกซ์โพเนนเชียล ฟังก์ชันลอการิทึม ฟังก์ชันตรีโกณมิติ ฟังก์ชันค่าสัมบูรณ์
ฟังก์ชันเวียนเกิด

