# ทฤษฏีที่เกี่ยวข้อง

ทฤษฏี กระบวนการและ เครื่องมือต่างๆ ที่เกี่ยวข้องในการจัดทำโปรแกรมแสดงผลข้อมูลชนิดกราฟ ออกมาในรูปแบบกราฟิก มีดังนี้

2.1 กราฟ และ ขั้นตอนวิธีของกราฟ

2.2 การควบคุมเวอร์ชันของรหัสโปรแกรมด้วย (SVN **S**ub**v**ersio**n)** และ Google Code

2.3 Python

2.4 Qt

2.5 Eclipse

ซึ่งจะอธิบายถึงรายละเอียด วิธีใช้ ตัวอย่าง และการนำมาใช้ของแต่ล่ะหัวข้อข้างต้นโดยละเอียดต่อไปนี้

## กราฟ และ ขั้นตอนวิธีของกราฟ

ในทางคณิตศาสตร์ และวิทยาการคอมพิวเตอร์ กราฟ คือวัตถุพื้นฐานของการศึกษาในทฤษฎีกราฟ กล่าวอย่างไม่เป็นทางการได้ว่า, กราฟคือเซตของวัตถุที่เรียกว่า จุดยอด, จุดตัด (vertex, node) ซึ่งเชื่อมต่อกันด้วย เส้นเชื่อม (edge) ในบางการประยุกต์ใช้งาน เส้นเชื่อมอาจแสดงอย่างมีทิศทางได้

**นิยาม**

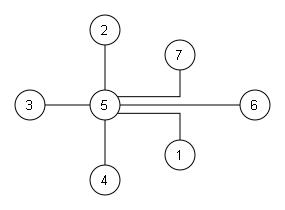
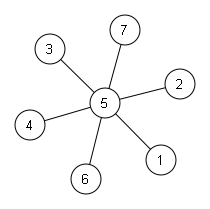
กราฟที่ไม่มีทิศทาง G = (V,E) คือโครงสร้างที่ประกอบด้วยกลุ่มของจุด V = {v1,v2,&,vm} และกลุ่มของเส้นเชื่อมระหว่างจุด 2 จุด E = {e1,e2, &, en} แต่ละเส้นเชื่อมจะต้องเกิดกับจุด 2 จุดที่ไม่เรียงลำดับ {m,v} จุด m และ v ไม่เป็นจุดเดียวกัน แต่ว่า {v,m} ถือว่าเป็นเส้นเดียวกันกับ {m,v}

graph

**รูปที่ 2-1** อธิบายความหมายกราฟ

ถ้ามีข้อมูลเป็นเซตของความสัมพันธ์ระหว่างเส้นเชื่อม และจุดตัด ชุดหนึ่งๆก็สามารถที่จะนำมาเขียนเป็นกราฟได้หลายรูปแบบในการแสดงผลแม้ว่า จะเป็นข้อมูลชุดเดียวกันก็ตาม

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| - | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 1 |  |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |  |
| 4 |  |  |  |  |  |  |  |
| 5 |  |  |  |  |  |  |  |
| 6 |  |  |  |  |  |  |  |
| 7 |  |  |  |  |  |  |  |

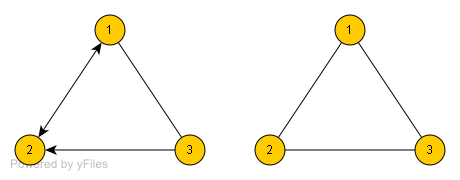


Organic graph layout Orthogonal graph layout adjacency matrix

**รูปที่ 2-2** ความแตกต่างของการแสดงผลกราฟ

**นิยาม**

กราฟที่มีทิศทาง G = (V, E) คือกราฟที่ประกอบด้วยกลุ่มของจุด V และกลุ่มของเส้นเชื่อม E โดยที่แต่ละเส้นเชื่อม จะเชื่อมจุด 2 จุด {u, v} ที่มีลำดับแน่นอน จากจุด u ไป v เส้นโยงดังกล่าวจะมีหัวลูกศรบอกทิศทาง จาก u ไป v โดยที่ u กับ v อาจจะเป็นจุดเดียวกันก็เป็นได้ หรือว่าจะมีการเชื่อมโยงกลับของv ไป u ก็เป็นได้

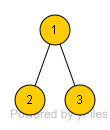
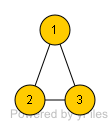


กราฟแบบมิทิศทาง กราฟแบบไม่มีทิศทาง

**รูปที่ 2-3** ความแตกต่างของกราฟแบบมีทิศทางและไม่มีทิศทาง

จากรูป 2-3 กราฟที่ เส้นเชื่อม มีทิศทางแม้เพียงเส้นเดียวก็ถือว่ากราฟนั้นเป็น กราฟที่มีทิศทางแล้ว ในที่นี้คือรูปตัวอย่างทางด้านซ้าย ส่วนทางด้านขวาคือกราฟไม่มีทิศทาง ซึ่งเรื่องของทิศทางนี้จะนำไปใช้พิจารณาในการจัดรูปแบบกราฟบางประเภท

กราฟยังแบ่งประเภทได้ด้วยการเชื่อมโยงของ เส้นเชื่อมได้อีกแบบคือ แบบต้นไม้(Tree) กับ แบบเครือข่าย (Network) โดย ต้นไม้คือกราฟที่ไม่เกิดการเชื่อมโยงเป็นวงขึ้นภายในตัวมันเองเลยและ แบบเครือข่ายคือกราฟที่มีการเชื่อมโยงกันเป็นวง ขึ้นภายในกราฟนั้นตั้งแต่ 1 วงขึ้นไป

ต้นไม้ เครือข่าย

**รูปที่ 2-4** ความแตกต่างของกราฟแบบต้นไม้และแบบเครือข่าย

### ขั้นตอนวิธีการหาต้นไม้แบบแผ่ที่เล็กที่สุด

**นิยาม**

ต้นไม้แบบแผ่ T ของกราฟ G = (V, E) คือต้นไม้ที่ประกอบด้วยจุดทุกจุดใน V และต้นไม้แบบแผ่ที่เล็กที่สุดคือต้นไม้แบบแผ่ที่มีผลรวมของน้ำหนักบนเส้นเชื่อมทั้งหมดน้อยที่สุด

**ขั้นตอนวิธีของ Prim**

เป็นวิธีในการหาต้นไม้แบบเล็กที่สุด โดยการสร้างต้นไม้แบบแผ่จากต้นเล็กๆ ก่อนแล้วค่อยๆ ขยายขนาดของต้นไม้จนกระทั่งมันคลุมจุดทุกจุดใน V ของกราฟ G = (V, E, W) วิธีของ Prim เป็นดังต่อไปนี้ ให้ VS เป็นที่เก็บกลุ่มของจุดสำหรับต้นไม้แบบแผ่ที่เล็กที่สุด และ T เป็นที่เก็บกลุ่มของเส้นเชื่อมของต้นไม้แบบแผ่ที่เล็กที่สุด โดยสามารถสรุปการทำงานได้เป็นขั้นตอนดังนี้

1. T = ∅; VS = ∅
2. เลือกจุดแรกตามใจชอบ สมมติ ว่าเลือก a;
3. V = V – {a}: VS = VS  ∪ {a};
4. while V ≠ ∅ do
   1. begin
      1. เลือกจุดที่อยู่ใน V ที่อยู่ติดกับจุดที่อยู่ใน VS โดยที่เส้นโยงระหว่างจุดที่เลือกกับจุดใน VS มีน้ำหนักน้อยที่สุด สมมติว่าจุดที่เลือกคือจุด U
      2. VS = VS ∪ {w}; T = TU {{b,w}} เมื่อ b คือจุดที่อยู่ติดกับ w   
         และ b ∈ VS; V = V – {w};
   2. end
5. end

ระยะเวลาที่ใช้ในการหาต้นไม้แบบแผ่โดยวิธีของ Prim เท่ากับ O(n2) เมื่อ n = ||V|| และ e = ||E||

### ขั้นตอนวิธีการหาระยะทางที่สั้นที่สุดระหว่างจุด 2 จุดในกราฟ

กราฟที่ใช้กับขั้นตอนวิธีนี้เป็นกราฟที่ไม่มีทิศทาง แต่ว่าเป็นกราฟที่มีน้ำหนักซึ่งน้ำหนักนี้ ใช้แทนระยะทางระหว่างจุด 2 จุด บนเส้นเชื่อมใดๆ ปัญหาของการหาระยะทางที่สั้นที่สั้นที่สุดระหว่างจุด 2 จุด พิจารณาได้โดยการสมมติการเดินระหว่างจุด a ไปจุด b โดยที่มีจุด x เชื่อมระหว่างเส้นทางจาก a ไป b และมีเส้นเชื่อม a ไป b โดยตรงด้วย ซึ่งจะพิจารณาว่าการระยะทาง a ไป x เป็น l(x) และ a ไป b เป็น l(b) และ w(x,b) คือระยะระหว่าง x และ b โดยกำหนดให้ x เป็นจุดหนึ่งใน Q ซึ่งอาจทำให้ l(x)+w(x,b) น้อยกว่า l(b) ดังนั้นวิธีการหาระยะทางที่สั้นที่สุดเป็นดังต่อไปนี้

กำหนดให้ V เป็นจุดเริ่มต้นทั้งหมด

1. เริ่มต้นให้ P = {a} และ Q = V – {a} สำหรับทุก ๆ จุด b ใน Q ให้ l(b) = w(a,b)
2. ให้ x เป็นจุดหนึ่งใน Q ซึ่งให้ค่า w(a,x) ต่ำสุดในบรรดาจุดทั้งหมดใน Q
3. ถ้าจุด x เป็นจุดที่ต้องการจุดสุดท้าย ให้หยุดการค้นหาระยะทาง ถ้าไม่ใช่ให้ P’ = P ∪ {x} และ Q’ = Q – {x} สำหรับทุกจุด b ให้ Q’ คำนวณหาระยะทางจากจุด x ไปยังจุด b ดังต่อไปนี้
4. l’(b) = min {l(b), l(x) + w(x ,b)}
5. ทำซ้ำในขั้นตอนที่ 2 โดยการแทน P ด้วย P’ และ Q ด้วย Q’ ค่าของ w(a, x) จะเท่ากับ **∞** ถ้าไม่มีเส้นเชื่อมระหว่างจุด a และจุด x

ระยะเวลาในการหาเส้นทางที่สุดในกราฟเท่ากับ O(n2) เมื่อ n = ||V||

## การควบคุมเวอร์ชันของรหัสโปรแกรมด้วย (SVN Subversion) และ Google Code

เนื่องจากขณะที่โปรแกรมยังอยู่ในขั้นตอนของการพัฒนานั้นจะมีการเปลี่ยนแปลงของ Source code ของโปรแกรมอย่างมาก และอาจจะมีการเปลี่ยนโครงสร้างของโปรแกรมด้วยซึ่งการแก้ไขบางอย่างอาจจะทำให้โปรแกรมเสียหายหรือไม่สามารถใช้งานได้ ซึ่งปัญหานี้สามารถแก้ได้โดยการทำ สำรองข้อมูลไฟล์(back up file) สำหรับแต่ล่ะรุ่นของโปรแกรม แต่ว่าการทำสำรองข้อมูลทั้งหมดของโปรแกรม ทุกๆเวอร์ชันจะทำให้เปลืองพื้นที่ในฮาร์ดดิสก์มาก ซึ่งปัจจุบันปัญหาแบบนี้ได้มีผู้ที่คิดวิธีในการแก้ไขปัญหาแบบนี้แล้วคือการใช้ ซอฟแวร์ในการควบคุมรหัสโปรแกรม(source code) โดยโปรแกรมที่เป็นที่รู้จักกันมากในปัจจุบันได้แก่ SVN (**S**ub**v**ersio**n**)และ CVS (Concurrent Versions System) ซึ่งในที่นี้ผู้พัฒนาได้เลือกใช้ SVN เพราะว่า SVN สนับสนุนการ เปลี่ยนชื่อไฟล์ และการลบไดเรคทอรี ซึ่งเป็นเรื่องสำคัญในช่วงแรกของการพัฒนาโปรแกรมที่การออกแบบยังไม่สมบูรณ์ และเลือก Server ที่ใช้เป็นที่เก็บข้อมูล ที่น่าเชื่อถืออย่าง Google code

SVN หรือ Subversion เป็นโปรแกรมโอเพนซอร์สที่ใช้ในการควบคุมรหัสโปรแกรม ที่ริเริ่มจัดทำโดยบริษัท CollabNet Inc. โดย SVN นี้พัฒนาต่อมาจากโปรแกรม CVS ที่มีมาก่อนหน้า และเพิ่มความสามารถต่างๆ เข้ามาจนปัจจุบันมีผู้ใช้เยอะขึ้นมาก และมีโปรแกรมที่สนับสนุนในการใช้มาก เช่น plug-in ของโปรแกรม Eclipse ทำให้การใช้ SVN ไม่ได้ยึดติดกับการใช้ส่วนติดต่อผู้ใช้แบบ Text mode เท่านั้น แต่ว่าสามารถใช้ GUI ต่างๆ เข้ามาช่วยได้

Google Code เป็นบริการแบบหนึ่งของ Google ที่มีไว้เพื่อสนับสนุนการทำงานของโปรแกรมเมอร์โอเพนซอร์ส ซึ่งมีบริการในการจัดการเกี่ยวกับโครงการที่นักพัฒนาต้องการสร้างอยู่หลายอย่างเช่น

* Wiki สำหรับอภิปรายเกี่ยวกับโครงการในกรณีที่ทำร่วมกันหลายคน หรือว่าเอาไว้ทำเอกสารออนไลน์เกี่ยวกับโครงการได้
* Download เป็นที่สำหรับเก็บเอกสาร โปรแกรมหรือสิ่งที่จำเป็นต่างสำหรับโครงการและกำหนดได้ว่าใครสามารถดาวน์โหลด ซึ่งอาจจะเป็นกลุ่มผู้พัฒนาหรือบุคคลทั่วไปก็ได้
* Issue การตามรุ่นต่างๆของโปรแกรมเช่นในแต่ละรุ่นมีการพัฒนามีการแก้ไขหรือเปลี่ยนแปลงอะไรบ้าง
* Source เป็นที่สำหรับบริการให้ดาวน์โหลดรหัสโปรแกรมซึ่งมีสองแบบคือ
  + กลุ่มผู้พัฒนา(มีการส่งซอร์สโค้ด ที่ได้ทำการแก้ไขกลับไปที่เซิร์ฟเวอร์ )
  + บุคคลทั่วไป(ไม่มีการส่งซอร์สโค้ดที่ได้ทำการแก้ไขกลับไปที่เซิร์ฟเวอร์ )

ซึ่งยังมีการให้บริการในการควบคุมรุ่นของโปรแกรมด้วย SVN ด้วยซึ่งผู้พัฒนาได้ใช้เซิร์ฟเวอร์  ของ Google นี้เป็นเซิร์ฟเวอร์  SVN สำหรับการพัฒนาโปรแกรมแสดงผลข้อมูลประเภทกราฟออกมาในรูปแบบกราฟิกด้วย

## Python

* **ข้อมูลเบื้องต้น**

ไพธอน (Python) เป็นภาษาโปรแกรมในลักษณะภาษาอินเตอร์พรีเตอร์โปรแกรมมิ่ง (Interpreted programming language) ผู้คิดค้นคือ Guido van Rossum ในปี 1990 ซึ่ง Python เป็น การจัดการชนิดของตัวแปรแบบแปรผันตามข้อมูลที่บรรจุอยู่ (Fully dynamically typed) และใช้การจัดการหน่วยความจำเป็นอัตโนมัติ (Automatic memory management) โดยได้เป็นการพัฒนาและผสมผสานของภาษาอื่น ๆ ได้แก่ ABC, Modula-3, Icon, ANSI C, Perl, Lisp, Smalltalk และ Tcl และภาษา Python ยังเป็นแนวคิดที่ทำให้เกิดภาษาใหม่ ๆ ซึ่งได้แก่ Ruby และ Boo เป็นต้น

Python นั้นพัฒนาเป็นโครงการ Open source โดยมีการจัดการแบบไม่หวังผลกำไรโดย Python Software Foundation และสามารถหาข้อมูลและตัวแปรภาษาได้จากเว็บไซต์ของ Python เอง ที่ http://www.python.org/ ซึ่งในปัจจุบัน (ณ.วันที่ 29 กันยายน 2551) Python ได้พัฒนาถึงรุ่นที่ 2.5.2 (ออกวันที่ 22 กุมภาพันธ์ 2551) และรุ่นทดสอบการทำงาน หรือ beta นั้นอยู่ที่รุ่น 2.6rc2(12 กันยายน 2551) และ 3.0rc1 (17 กันยายน 2551)

* **Python 1.0**

Python สร้างขึ้นครั้งแรกในปี 1990 โดย Guido van Rossum ที่ CWI (National Research Institute for Mathematics and Computer Science) ในประเทศเนเธอร์แลนด์ โดยได้นำความสำเร็จของภาษาโปรแกรมมิ่งที่ชื่อ ABC มาปรับใช้กับ Modula-3, Icon, C, Perl, Lisp, Smalltalk และ Tcl โดย Guido van Rossum ถือว่าเป็นผู้ริเริ่มและคิดค้น แต่เค้าก็ยังคิดว่าผลงานอย่าง Python นั้น เป็นผลงานความรู้ที่ทำขึ้นเพื่อความสนุกสนานโดยได้อ้างอิงงานชิ้นนี้ของเขาว่าเป็น Benevolent Dictator for Life (BDFL) ซึ่งผลงานที่ถูกเรียกว่าเกิดจากความสนุกสนานเหล่านี้นั้นมักถูกเรียกว่า BDFL เพราะมักเกิดจากความไม่ตั้งใจ และความอยากที่จะทำอะไรที่เป็นอิสระนั้นเอง ซึ่งคนที่ถูกกล่าวถึงว่าทำในลักษณะแบบนี้ก็ได้แก่ Linus Torvalds ผู้สร้าง Linux kernel, Larry Wall ผู้สร้าง Perl programming language และคนอื่น ๆ อีกมากมาย

โดยที่ใน Python 1.2 นั้นได้ถูกปล่อยออกมาในปี 1995 โดย Guido ได้กลับมาพัฒนา Python ต่อที่ Corporation for National Research Initiatives (CNRI) ที่ เรสตัน, มลรัฐเวอร์จิเนีย ประเทศสหรัฐอเมริกา โดยที่ในขณะเดียวกันก็ได้ปล่อยรุ่นใหม่ ในหมายเลขรุ่น 1.6 ออกมาโดยอยู่ที่ CNRI เช่นกัน

ซึ่งหลังจากปล่อยรุ่น 1.6 ออกมาแล้ว Guido van Rossum ก็ได้ออกจาก CNRI  เพื่อทำงานให้การทำธุรกิจพัฒนาซอฟต์แวร์แบบเต็มตัว โดยก่อนที่จะเริ่มทำงานธุรกิจ เขาก็ได้ทำให้ Python นั้นอยู่บนสัญญาลิขสิทธิ์แบบ General Public License (GPL) โดยที่ CNRI และ Free Software Foundation (FSF) ได้รวมกันเปิดเผยรหัสโปรแกรมทั้งหมด เพื่อให้ Python นั้นได้ชื่อว่าเป็นซอฟต์แวร์เสรี และเพื่อให้ตรงตามข้อกำหนดของ GPL-compatible ด้วย (แต่ยังคงไม่สมบูรณ์เพราะการพัฒนาในรุ่น 1.6 นั้นออกมาก่อนที่จะใช้สัญญาลิขสิทธิ์แบบ GPL ทำให้ยังมีบางส่วนที่ยังเปิดเผยไม่ได้)

และในปีเดียวกันนั้นเอง Guido van Russum ก็ได้รับรางวัลจาก FSF ในชื่อว่า "Advancement of Free Software"

โดยในปีนั้นเอง Python 1.6.1 ก็ได้ออกมาเพื่อแก้ปัญหาข้อผิดพลาดของตัวซอฟต์แวร์และให้เป็นไปตามข้อกำหนดของ GPL-compatible license อย่างสมบูรณ์

* **Python 2.0**

ในปี 2000 Guido และ Python Core Development team ได้ย้ายการทำงานไป BeOpen.com โดยที่พวกเขาได้ย้ายจาก BeOpen PythonLabs team โดยใน Python รุ่นที่ 2.0 นั้นได้ถูกนำออกเผยแพร่ต่อบุคคลทั่วไปจากเว็บไซต์ BeOpen.com และหลังจากที่ Python ออกรุ่นที่ 2.0 ที่ BeOpen.com แล้ว Guido และนักพัฒนาคนอื่น ๆ ในทีม PythonLabs ก็ได้เข้ารวมกับทีมงาน Digital Creations

Python รุ่น 2.1 ได้สืบทอดการทำงานและพัฒนามาจาก 1.6.1 มากกว่า Python รุ่น 2.0 และได้ทำการเปลี่ยนชื่อสัญญาลิขสิทธิ์ใหม่เป็น Python Software Foundation License โดยที่ใน Python รุ่น 2.1 alpha นั้นก็ได้เริ่มชื่อสัญญาลิขสิทธิ์นี้และผู้เป็นเจ้าของคือ Python Software Foundation (PSF) โดยที่เป็นองค์กรที่ไม่หวังผลกำไรเช่นเดียวกับ Apache Software Foundation

* **อนาคต**

ผู้พัฒนา Python มีการประชุมและถกเถียงกันในเรื่องของความสามารถใหม่ ๆ ใน Python รุ่นที่ 3.0 โดยมีชื่อโครงการว่า Python 3000 (Py3K) โดยที่จะหยุดการสนับสนุนโค้ดโปรแกรมจากรุ่น 2.x โดยที่ทำแบบนี้เพื่อทำการปรับปรุงเปลี่ยนแปลงการทำงานของภาษาให้ดียิ่งขึ้นตามคำแนะนำที่ว่า "reduce feature duplication by removing old ways of doing things" (ลดทอนคุณสมบัติที่ซ้ำซ้อนด้วยการยกเลิกเส้นทางที่เดินผ่านมาแล้ว) โดยในตอนนี้ยังไม่มีตารางงานของ Python รุ่น 3.0 แต่อย่างใด แต่ Python Enhancement Proposal (PEP) ได้มีการวางแผนไว้แล้ว โดยได้วางแผนไว้ดังนี้

* + เพิ่มส่วนสนับสนุนชนิดตัวแปรให้มากขึ้น
  + สนับสนุนการทำงานของชนิดตัวแปรแบบ unicode/str และ separate mutable bytes type
  + ยกเลิกการสนับสนุนคุณสมบัติของ classic class, classic division, string exceptions และ implicit relative imports
  + ฯลฯ
* **หลักปรัชญาของภาษาไพธอน**

Python เป็นภาษาที่สามารถสร้างงานได้หลากหลายกระบวนทัศน์ (Multi-paradigm language) โดยจะมองอะไรที่มากกว่าการ coding เพื่อนำมาใช้งานตามรูปแบบเดิม ๆ แต่จะเป็นการนำเอาหลักการของกระบวนทัศน์ (Paradigm) แบบ Object-oriented programming, Structured programming, Functional programming และ Aspect-oriented programming นำเอามาใช้ทั้งแบบเดี่ยว ๆ และนำมาใช้ร่วมกัน ซึ่ง Python นั้นเป็น ภาษาที่มีการตรวจสอบชนิดตัวแปรแบบยืดหยุ่น (dynamically type-checked) และใช้ Garbage collection ในการจัดการหน่วยความจำ

* **ข้อเด่นของภาษาไพธอน**
* ง่ายต่อการเรียนรู้ โดยภาษา Python มีโครงสร้างของภาษาไม่ซับซ้อนเข้าใจง่าย ซึ่งโครงสร้างภาษา Python จะคล้ายกับภาษาซีมาก เพราะภาษา Python สร้างขึ้นมาโดยใช้ภาษาซี ทำให้ผู้ที่คุ้นเคยภาษาซี อยู่แล้วใช้งานภาษา Python ได้ไม่ยาก นอกจากนี้โดยตัวภาษาเองมีความยืดหยุ่นสูงทำให้การจัดการกับงานด้านข้อความ และ Text File ได้เป็นอย่างดี
* ไม่ต้องเสียค่าใช้จ่ายใดๆ ทั้งสิ้น เพราะตัวแปรภาษา Python อยู่ภายใต้ลิขสิทธิ์ Python Software Foundation License (PSFL) ซึ่งเป็นของ Python Software Foundation (PSF) ซึ่งมีลักษณะคล้ายกับ ลิขสิทธิ์แม่แบบอย่าง General Public License (GPL) ของ Free Software Foundation (FSF)
* ใช้ได้หลายแพลตฟอร์ม ในช่วงแรกภาษา Python ถูกออกแบบใช้งานกับระบบ Unix อยู่ก็จริง แต่ในปัจจุบันได้มีการพัฒนาตัวแปลภาษา Python ให้สามารถใช้กับระบบปฏิบัติการอื่นๆ อาทิเช่น Linux Platform, Windows Platform, OS/2, Amiga, Mac OS X และรวมไปถึงระบบปฎิบัติการที่ .NET Framework, Java virtual machine ทำงานได้ ซึ่งใน Nokia Series 60 ก็สามารถทำงานได้เช่นกัน
* ภาษา Python ถูกสร้างขึ้นโดยได้รวบรวมเอาส่วนดีของภาษาต่างๆ เข้ามาไว้ด้วยกัน อาทิเช่น ภาษา ABC, Modula-3, Icon, ANSI C, Perl, Lisp, Smalltalk และ Tcl
* Python สามารถรวมการพัฒนาของระบบเข้ากับ COM, .NETและ CORBA objects
* สำหรับ Java libraries แล้วสามารถใช้ Jython เพื่อทำการพัฒนาซอฟต์แวร์จากภาษา Python สำหรับ Java Virtual Machine
* สำหรับ .NET Platform แล้ว สามารถใช้ IronPython ซึ่งเป็นการพัฒนาของ Microsoft เพื่อจะทำให้ Python นั้นสามารถทำงานได้บน .Net Framework ซึ่งใช้ชื่อว่า Python for .NET
* Python นั้นสนับสนุน Internet Communications Engine (ICE) และการรวมกันของเทคโนโลยีอื่น ๆ อีกมากมายในอนาคต
* บางครั้งนักพัฒนาอาจจะพบว่า Python ไม่สามารถทำงานบางอย่างได้ แต่นักพัฒนาต้องการให้มันทำงานได้ ก็สามารถพัฒนาเพิ่มได้ในรูปแบบของ extension modules ซึ่งอยู่ในรูปแบบของโค้ด C หรือ C++ หรือใช้ SWIG หรือ Broost.Python
* ภาษา Python เป็นภาษาที่สามารถพัฒนาโปรแกรมประเภท Server side Script คือการทำงานของภาษา Python จะทำงานด้านฝั่ง Server แล้วส่งผลลัพธ์กลับมายัง Client ทำให้มีความปลอดภัยสูง และยังใช้ภาษา Python นำมาพัฒนาเว็บเซอร์วิสได้อีกด้วย
* ใช้พัฒนาระบบบริหารการสร้างเว็บไซต์สำเร็จรูปที่เรียกว่า Content Management Systems (CMS) ซึ่ง CMS ที่มีชื่อเสียงมาก และเบื้องหลังทำงานด้วย Python คือ Plone http://www.plone.org/
* **Category และ Application Domains**

ภาษา Python นั้น จัดอยู่ใน Category ภาษาที่สามารถสร้างงานได้หลากหลายกระบวนทัศน์ (Multi-paradigm language) โดยรองรับทั้ง Object-oriented programming, Imperative, Functional programming และ Logic programming ซึ่ง Python สามารถนำไปพัฒนาซอฟต์แวร์ประยุกต์ได้มากมาย ได้แก่

* **Web และ Internet Development**

Python นั้นมีการสนับสนุนในด้านของ Web Development ในโซลูชันระดับสูงด้วย Zope, mega frameworks อย่าง Django และ TurboGears และรวมไปถึง Content Management Systems ขั้นสูงอย่าง Plone และ CPS จึงทำให้ Python นั้นเป็น Common Gateway Interface (CGI) ระดับสูงที่มีประสิทธิภาพที่ดีที่สุดตัวหนึ่งในตลาด

* **Database Access**

Python นั้นสนับสนุนการเข้าถึงข้อมูลในฐานข้อมูลของผู้ผลิตฐานข้อมูลต่าง ๆ มากมาย โดยผ่านทาง ODBC Interfaces และ Database Connection Interface อื่น ๆ ซึ่งสามารถทำงานร่วมกับ  MySQL, Oracle, MS SQL Server, PostgreSQL, SybODBC และอื่น ๆ ที่จะมีมาเพิ่มเติมอีกในอนาคต

* **Desktop GUIs**

เมื่อ Python ได้ติดตั้งลงบนเครื่องของคุณแล้ว จะมี Tk GUI development library ซึ่งเป็น libraries ที่มีความสามารถเทียบเท่า Microsoft Foundation Classes (MFC, ซึ่งคล้าย ๆ กับ win32 extensions), wxWidgets, GTK, Qt, Delphi และอื่น ๆ ทำให้สามารถพัฒนาซอฟต์แวร์ประยุกต์ต่าง ๆ แบบ Graphic user interface ได้

* **Scientific และ Numeric computation**

Python รองรับการทำงานของนักวิทยาศาสตร์ในเรื่องของทฤษฎีการคำนวณ, Bioinformatics และ Physics เป็นต้น

* **Education**

Python นั้นเป็นภาษาที่เหมาะกับการเรียนการสอนในวิชา programming อย่างมาก โดยสามารถนำไปใช้ในระดับเบื้องต้นถึงระดับสูง ซึ่ง Python Software Foundation นั้นได้มีหลักสูตรสำหรับการเรียนการสอนในด้านนี้อยู่แล้ว ซึ่งสามารถนำเอา pyBiblio  และ Software Carpentry Course มาเรียนเพื่อเสริมความรู้ได้

* **Network programming**

เป็นการเพิ่มความสามารถจาก Web และ Internet Development Python นั้นสนับสนุนในการเขียนโปรแกรมในระดับต่ำในด้านของ network programming ที่ง่ายต่อการพัฒนา sockets และ รวมไปถึงการทำงานร่วมกับ modules อย่าง Twisted และ Framework สำหรับ Asynchronous network programming

* **Software build และ Testing**

Python นั้นสนับสนุนการพัฒนาซอฟต์แวร์ที่มีการควบคุมการพัฒนาและจัดการระบบทดสอบต่าง ๆ โดยใช้เครื่องมือในการพัฒนาที่สนับสนุนการเขียนโปรแกรมใน Python เอง ซึ่งตัว Python นั้นได้มาพร้อมกับ

* Scons สำหรับ build โปรแกรม
* Buildbot และ Apache Gump ที่ใช้สำหรับงาน Automated continuous compilation และ Testing
* Roundup หรือ Track สำหรับ bug tracking และ project management
* **Game และ 3D Graphics Rendering**

Python นั้นได้ถูกใช้ในตลาดพัฒนาเกมส์ทั้งเชิงธุรกิจและสมัครเล่น โดยมีการสร้าง Framework สำหรับพัฒนา Game บน Python ซึ่งชื่อว่า PyGame และ PyKyra ซึ่งยังรวมไปถึงการทำ 3D Graphics Rendering ที่ Python มี libraries ทางด้านงานนี้อยู่มากมาย

* **ซอฟต์แวร์ที่เขียนด้วยไพธอน**
* **BitTorrent**เป็นการพัฒนาโดยระบบการจัดการไฟล์ BitTorrent, การจัดการ การกระจายตัวของ Package ข้อมูลใน Tracker และการเข้ารหัสส่วนข้อมูลต่าง ๆ
* **Blender**ซอฟต์แวร์ open source สำหรับทำ 3D modeling
* **Chandler**ซอฟต์แวร์จัดการข้อมูลส่วนบุคคล (Personal Information Manager, PIM) โดยมีส่วนเพิ่มเติมทั้งงานปฎิทิน, อีเมล, ตารางงาน และข้อมูลโน็ตต่าง ๆ ซึ่งทำงานคล้าย ๆ กับ Outlook ของ Microsoft
* **Civilization IV** วีดิโอเกมส์ และยังเป็นเกมส์ที่ใช้ boost.python เพื่อทำการควบคุมส่วนประกอบต่าง ๆ ภายในเกมส์ ซึ่งรวมไปถึงรูปแบบ, หน้าตา และเนื้อหาของเกมส์ด้วย
* **Mailman**หนึ่งในซอฟต์แวร์ E-Mail mailing lists ที่ได้รับความนิยมสูงสุด
* **Kombilo**ระบบจัดการฐานข้อมูลของเกมส์โกะ
* **MoinMoin**ระบบ Wiki ที่ได้รับความนิยมสูงตัวหนึ่ง
* **OpenRPG**ระบบเกมส์เสมือนแบบ Role Playing Games บน Internet
* **Plone**ระบบ Content Management System
* **Trac**ระบบติดตามติดตามข้อผิดพลาดและจัดการข้อมูลด้านการพัฒนาซอฟต์แวร์ด้วย MoinMoin ที่เป็น wiki และ Subversion เพื่อทำระบบ Source version control
* **Turbogears**ระบบพัฒนาซอฟต์แวร์ Framework โดยรวมเอา Cherrypy, SQLObject, MochiKit และ KID templates
* **ViewVC**ระบบ Web-based สำหรับจัดการด้าน CVS และ SVN repositories
* **Zope**ระบบพัฒนาซอฟต์แวร์บนอินเทอร์เน็ตแบบ web-application platform
* **Battlefield 2** เกมส์ First Person Shooter ที่ได้ใช้ Python ในการทำ Configuration scripts
* **Indian Ocean Tsunami Detector**ซอฟต์แวร์สำหรับมือถือเพื่อแจ้งเตือน Tsunami
* **EVE Online** เกมส์แบบ Multi Massive Online Role Playing Game ซึ่งเป็นเกมส์ที่ได้รับอันดับสูงมากบน MMORPG.com
* **SPE - Stani’s Python Editor**เป็น Free และ open-source สำหรับงานพัฒนาซอฟต์แวร์ด้วย Python โดยมีทั้งแบบ Python IDE for Windows, Linux & Mac with wxGlade (GUI designer), PyChecker (Code Doctor) และ Blender (3D)

## Qt

Qt (อ่านว่า cute) เป็น cross-platform framework ในการพัฒนาโปรแกรมที่ถูกพัฒนาขึ้นโดย Trolltech เพื่อใช้ในการพัฒนาโปรแกรมในรูปแบบกราฟิกหรือเป็นที่รู้จักกันในอีกชื่อว่า widget-toolkit แล้วก็ยังถูกใช้ในพัฒนาโปรแกรมที่เป็นแบบ Text mode หรือว่าโปรแกรมที่ทำงานอยู่บนเซิร์ฟเวอร์ก็ได้ Qt ถูกใช้อย่างเห็นได้ชัดในหลายโปรแกรมเช่น KDE(Linux Desktop Environment) ,Opera (Web browser),Google Earth, Skype, Photoshop Element ,VirtualBox ,VLC Media Playerเป็นต้น

Qt สร้างขึ้นมาโดยใช้ C++ หลายส่วนขยายของโปรแกรมที่ไม่เป็นมาตรฐานถูกเขียนขึ้นโดยเพิ่มเข้ามากับมาโค้ดที่เป็นมาตรฐานของ C++ ก่อนที่จะ compile ออกมาเป็น Qt โดยเป้าหมายการการพัฒนาโปรแกรมของ Qt มีแนวความคิดมากจากคำว่า

*“Code less Create more Deploy everywhere”*

*“เขียนโค้ดน้อยลง สร้างสรรค์ผลงานได้มากขึ้น นำไปใช้ได้ทุกที่”*

ดังนั้น Qt แม้ว่าจะเขียนขึ้นมาด้วย C++ แต่ว่าก็สามารถที่นำไปใช้กับภาษาโปรแกรมภาษาอื่นได้ เช่น Ada (QtAda), C# (Qyoto/Kimono),Java(Qt Jambi), Pascal, Perl, PHP(PHP-Qt),Ruby( RubyQt) และภาษาที่ผู้พัฒนาได้เลือกมาใช้กับ framework นี้คือ Python(PyQt) ซึ่งทั้งตัวภาษาและ framework สามารถทำงานได้บน platforms หลักๆ ได้ทั้งหมดและยังสนับสนุนเรื่องภาษาท้องถิ่นจำนวนมากรวมถึงภาษาไทย มีความสามารถอื่นๆที่ไม่ใช่ด้านกราฟิกเพื่อสนับสนุนการเขียนโปรแกรมอีกได้แก่ SQL, XML, การจัดการกับการทำงานแบบ Thread ,Network ,และการจัดการไฟล์ข้าม platform

Qt ถูกพัฒนาออกมา 4 Editions สำหรับจุดประสงค์ต่างกันคือ

* Qt Console Edition สำหรับพัฒนาโปรแกรมแบบคอนโซลคือเป็น Text mode หรือโปรแกรมที่ทำงานเบื้องหลัง
* Qt Desktop Light สำหรับเขียนโปรแกรม GUI ที่มีขนาดเล็ก โดยตัดความสามารถเรื่อง network และ database ออกไป
* Qt Desktop Edition สมบูรณ์ที่มีทุกความสามารถครบ
* Qt Open Source Edition สมบูรณ์ที่มีข้อยกเว้นเกี่ยวกับเรื่องสัญญาอนุญาตของ ซอฟต์แวร์ (ซึ่งเป็นรุ่นที่ผู้พัฒนาเลือกนำมาใช้ในการพัฒนาโปรแกรม)

Qt ถูกพัฒนาขึ้นมาแล้วทั้งหมด 4 รุ่นซึ่งรุ่นล่าสุดคือรุ่นที่ 4 ซึ่งมีความสามารถหลักๆเพิ่มขึ้นมาอีก 4 อย่างคือ

* เซตของ Template
* Interviews เพิ่มหลักการของ model/views ให้กับวัตถุที่ใช้ในการแสดงผล(item views)
* Arthur framework สำหรับการสร้างรูปสองมิติ
* Scribe การ render ข้อความที่เข้ารหัสแบบ Unicode
* MainWindow ส่วนเสริมที่ในการสร้างหน้ากากหลักของโปรแกรม

ซึ่งในรุ่นที่ 4 ของ Qt ก็ได้ออกรุ่นย่อยออกมา 4 รุ่นซึ่งแต่ล่ะรุ่นก็มีความสามารถใหม่ๆเพิ่มขึ้นมาซึ่งผู้พัฒนาได้เลือกรุ่นล่าสุดมาใช้คือรุ่น 4.4 เนื่องจากก่อนหน้านี้ยังไม่สนับสนุน Vista และรุ่นนี้ก็เพิ่มความสามารถแล้วแก้ไขปัญหาของรุ่นก่อนหน้าเรียบร้อย

### PyQt4

เซตของ python แปลงมาจาก Qt4 เพื่อให้สามารถนำมาเขียนได้ง่ายในรูปแบบภาษาแบบ python ซึ่งได้รวมเอาความสามารถต่างๆ ของ Qt4 ไว้ครบถ้วนซึ่งผู้ใช้ไม่ต้องเรียนรู้ Syntax ของภาษาใหม่แต่อย่างใด แต่ต้องเรียนรู้วิธีการใช้ และหลักการที่เราจะนำมาใช้ในการพัฒนาโปรแกรมซึ่งทำให้ลดเวลาในการปรับตัวลงได้มาก และทำให้เราสามารถสร้างงานได้มากขึ้น

## Eclipse

Eclipse เป็นเครื่องมือที่ช่วยสำหรับในการพัฒนาโปรแกรมสำหรับนักพัฒนาโปรแกรม Java เป็นหลัก โดยตัวโปรแกรมเองเป็นซอฟต์แวร์โอเพนซอร์ส ซึ่งปัจจุบันได้มีคนจำนวนมากได้ช่วยกันพัฒนาโปรแกรมตัวนี้ให้สามารถพัฒนาโปรแกรมภาษาอื่นเพิ่มขึ้นได้อีกเป็นจำนวนมากและมีการสร้างเครื่องมือเสริมอื่นๆขึ้นมาอีกเป็นจำนวนมากเช่น การควบคุมรุ่นของโปรแกรมด้วย SVN, CVSเครื่องมือการเขียนโปรแกรมภาษาอื่นๆ Pydev(ภาษา Python ), RadRail (Ruby on rail), PHP, C/C++ เครื่องมือเกี่ยวกับการพัฒนาเว็บเช่น browser ภายในตัวเอง, XML Editor, Database tool เครื่องมือในการวัดประสิทธิภาพของโปรแกรม profiler, Enerjy

อย่างไรก็ตามเครื่องมือเสริมต่างๆ ข้างต้นถ้าเราต้องการใช้เราต้องติดตั้งเพิ่มขึ้นเอง แต่ก็มีกลุ่มของผู้พัฒนาที่ใช้งานบางอย่างเหมือนๆ กันเช่น กลุ่มของนักพัฒนาเว็บ กลุ่มของนักพัฒนาโปรแกรมด้วยภาษาต่างๆ กลุ่มที่พัฒนาโปรแกรมเฉพาะทาง จึงได้มีกลุ่มที่คิดที่จะทำ เครื่องมือเสริมของ Eclipse รวมกับ Eclipse ไว้เป็นชุดๆ เพื่อความง่ายในการเฉพาะทางโดนไม่ต้องมาเสียเวลาในการลงเครื่องมือเสริมเหล่านั้นใหม่ โดยเรียก Eclipse แบบนี้ว่า EasyEclipse โดยชุดโปรแกรมของEasyEclipse มีหลายรูปแบบเช่น

* EasyEclipse Expert Java สำหรับผู้ที่เขียนโปรแกรม Java แบบมืออาชีพ
* EasyEclipse Desktop Java สำหรับผู้ที่พัฒนาโปรแกรม Java แบบเน้นไปที่โปรแกรมบน Desktop
* EasyEclipse Server Java สำหรับผู้ที่พัฒนาโปรแกรม Java แบบเน้นไปที่โปรแกรมด้าน Server
* EasyEclipse Mobile Java สำหรับผู้ที่พัฒนาโปรแกรม Java แบบเน้นไปที่โปรแกรมบนมือถือหรืออุปกรณ์ประเภท Mobile device ต่างๆ
* EasyEclipse for LAMP สำหรับผู้พัฒนาโปรแกรมบน LAMP
* EasyEclipse for PHP สำหรับผู้พัฒนาเว็บด้วย PHP
* EasyEclipse for Ruby and Rails สำหรับผู้พัฒนาเว็บด้วย Ruby on rail.
* EasyEclipse for C and C++ สำหรับผู้พัฒนาโปรแกรมด้วย C/C++

ในที่นี้ผู้พัฒนาได้เลือก EasyEclipse for LAMP มาใช้เป็นเครื่องมือสำหรับการพัฒนาเพราะว่า EasyEclipse for LAMP มีเครื่องมือที่ช่วยในการพัฒนาหลายอย่างรวมมาให้อยู่แล้ว ซึ่งเครื่องมือที่ผู้พัฒนาใช้หลักๆ ได้แก่ Pydev และ SVN ซึ่ง EasyEclipse for LAMP ยังมีความสามารถในการพัฒนา Ruby on rail, PHP และ Java อยู่ด้วย ทำให้ไม่จำเป็นต้องเปลี่ยนเครื่องมือบ่อยในการที่จะพัฒนาโปรแกรมต่างๆ ในแต่ละครั้ง

## GraphML

GraphML เป็น xml ชนิดหนึ่งซึ่งออกแบบมาเพื่อการเก็บข้อมูลชนิดกราฟ ซึ่งออกแบบโดย Graphdrawing.org โดยการออกแบบนั้นครอบคลุมทุกเนื้อหา และง่ายที่จะนำมาใช้กับข้อมูลประเภทกราฟ ซึ่งมันประกอบไปด้วยส่วนประกอบหลักของตัวภาษาที่ใช้ในการอธิบายข้อมูลประเภทกราฟ และยืดหยุ่นพอสำหรับที่จะให้มีส่วนเสริม เพิ่มเข้าไปสำหรับข้อมูลเฉพาะของแต่ละโปรแกรมที่นำไปใช้ ซึ่งส่วนประกอบหลักของตัวภาษาจะประกอบไปด้วย

* กราฟแบบมีทิศทาง กราฟไม่มีทิศทาง กราฟแบบผสม
* Hypergraphs
* Hierarchical graphs
* Graphical representations
* References to external data
* Application-specific attribute data
* Light-weight parsers

ซึ่ง GraphML ได้ถูกนำไปใช้ในหลายๆ โปรแกรมที่ใช้ข้อมูลประเภทกราฟ และมีอีกหลายๆโปรแกรมที่ให้การสนับสนุนการใช้ GraphML เช่น JUNG, yFile, AT&T Research, IBM Research ซึ่งผู้พัฒนาได้นำเอาไฟล์ประเภทนี้มาใช้ในการเก็บข้อมูลประเภทกราฟ แต่ว่าได้นำความสามารถของ GraphML มาแค่บางส่วนเท่านั้น