

## การออกแบบภาษา Hunnaeee

Token	Regular Expression	ตัวอย่าง
<b>ค่าคงที่</b>		
จำนวนเต็มฐาน 10	<code>[-]?[0-9]+</code>	-1 , -500 , 0 , 10 , 1000
จำนวนเต็มฐาน 16	<code>0[xX][0-9a-fA-F]+</code>	0x0000 , 0xAAA , 0xE4
<b>ตัวแปรพื้นฐาน</b>		
จำนวนเต็มขีดเครื่องหมายขนาด 64 บิต	<code>[a-z]{2}</code>	ab , bz , er
<b>นิพจน์คำนวณจำนวนเต็ม</b>		
ติดลบ	<code>"_"</code>	-aa , -5 , -0
คูณ	<code>"*"</code>	aa * bb , 0 * 5 , 10 * qq
หารเอาส่วน	<code>"/"</code>	aa / bb , 11 / 5 , 10 / qq
หารเอาเศษ	<code>"%"</code>	aa % bb , 11 % 5 , 10 % qq
บวก	<code>"+"</code>	aa + bb , 11 + 5 , 10 + qq
ลบ	<code>"_"</code>	aa - bb , 11 - 5 , 10 - qq
วงเล็บ	<code>"(" , ")"</code>	(5*8)+(3-1)
<b>ประโยคคำสั่งที่ทำตามลำดับ</b>		
ให้ค่าแก่ตัวแปร (assignment)	<code>"="</code>	aa = 5
แสดงค่าตัวแปรโตดแบบฐาน 10	<code>"print"</code>	print 5
แสดงค่าตัวแปรโตดแบบฐาน 16	<code>"print"</code>	print hex(78)
แสดงสายอักขรที่กำหนด (ยาวไม่เกิน 255 ตัว)	<code>"print"</code>	print "..."
<b>ประโยคคำสั่งตัดสินใจ</b>		
เท่ากัน	<code>"=="</code>	0 == 0
มากกว่าหรือเท่ากับ	<code>"&gt;="</code>	5 >= 5 , 5 >= 0
น้อยกว่าหรือเท่ากับ	<code>"&lt;="</code>	1 <= 10 , 10 <= 10
มากกว่า	<code>"&gt;"</code>	5 > 0
น้อยกว่า	<code>"&lt;"</code>	1 < 10
ประโยคเงื่อนไข แบบที่ 1	<code>"if"</code>	if pm == 0 { pr[id] = fm

		id = id + 1 }
ประโยคเงื่อนไข แบบที่ 2	“else”	else fm % st == 0 { pm = 1 }
ประโยคคำสั่งวนซ้ำ		
For loop	“for”	for fm : to { pm = 0; st = 2 }