

□ Structure de contrôle : *Si ... Alors ... Sinon ... FinSi*

<pre>Si (c > 4) Alors action1 Sinon action2 FinSi</pre>	<pre> graph TD A{condition C>4?} -- VRAIE --> B[ACTION 1] A -- FAUSSE --> C[ACTION 2] B -- suite --> D[suite] C -- suite --> D </pre>	Langage C <pre>if (c > 4) { Action1; } else { Action2; }</pre>
--	--	--

□ Structure de contrôle : *Tantque ... faire ... FinTantque*

<pre>Tantque (b = 1) Faire action FinFaire</pre>	<pre> graph TD A{b=1 ?} -- OUI --> B[ACTION 1] B -- suite --> C[suite] A -- NON --> C </pre>	Langage C <pre>While (b == 1) { action; }</pre> <p>(Test en tête de boucle)</p>
---	---	--

□ Structure de contrôle : *Faire ... Tantque ...*

<pre>Faire action Tantque (b = 0)</pre>	<pre> graph TD A[ACTION 1] --> B{b=0 ?} B -- OUI --> A B -- NON --> C[suite] </pre>	Langage C <pre>do { action; } while (b == 0); </pre> <p>(Test en fin de boucle)</p>
---	--	---

□ Structure de contrôle : *Pour ... variant de ... à ... par pas de ... Faire ... FinPour*

<pre>Pour i de 0 à 4 Faire action FinFaire</pre>	<pre> graph TD A[Charger valeur initiale V ← Vi] --> B{V = Vf ?} B -- OUI --> C[ACTION] C --> B B -- NON --> D[V ← V +/- pas] D --> E[suite] </pre> <p>V: variable Vi: valeur initiale de V Vf: Valeur Finale de V Pas: incrément ou décrément</p>	Langage C <pre>int i; for (i=0; i <= 4; i++) { action; }</pre>
---	---	---