STEM in the french high school curriculum
A teacher's challenge ?

Robert Cabane

Honorary general inspector of the french MoE

March 2019

How the system changes (before 2012)

<table>
<tr><th>Levels</th><th>School years</th><th>French names</th><th colspan="6">Particularities</th></tr>
<tr><td rowspan="6">High school (Lycée)</td><td rowspan="2">12</td><td rowspan="2">Terminale</td><td>Math</td><td>Physics</td><td>Biology</td><td>Engineering</td><td rowspan="2">Techno</td><td rowspan="6">Vocational schools</td></tr>
<tr><td colspan="4">Common courses (math, physics-chemistry, life sciences, …)</td></tr>
<tr><td rowspan="2">11</td><td rowspan="2">Première</td><td colspan="4">Science common courses (math, physics-chemistry, life sciences)</td><td rowspan="2">Techno</td></tr>
<tr><td colspan="4">Other common courses (french, history & geography…)</td></tr>
<tr><td rowspan="2">10</td><td rowspan="2">Seconde</td><td colspan="5">Science common courses (math, physics-chemistry, life sciences)</td></tr>
<tr><td colspan="4">Other common courses</td></tr>
<tr><td rowspan="4">Middle school (Collège)</td><td>9</td><td>Troisième</td><td colspan="6"></td></tr>
<tr><td>8</td><td>Quatrième</td><td colspan="6"></td></tr>
<tr><td>7</td><td>Cinquième</td><td colspan="6"></td></tr>
<tr><td>6</td><td>Sixième</td><td colspan="6"></td></tr>
</table>

How the system changes (2012-2016)

<table>
<tr><th>Levels</th><th>School years</th><th>French names</th><th colspan="7">Particularities</th></tr>
<tr><td rowspan="4">High school (Lycée)</td><td>12</td><td>Terminale</td><td>Math</td><td>Physics</td><td>Biology</td><td>ISN</td><td>Engineering</td><td>Techno</td><td rowspan="4">Vocational schools</td></tr>
<tr><td colspan="5">Common courses (math, physics-chemistry, life sciences, …)</td></tr>
<tr><td>11</td><td>Première</td><td colspan="5">Science common courses (math, physics-chemistry, life sciences)</td><td rowspan="2">Techno</td></tr>
<tr><td colspan="5">Other common courses (french, history & geography…)</td></tr>
<tr><td rowspan="3">10</td><td rowspan="3">Seconde</td><td colspan="6">Science common courses (math, physics-chemistry, life sciences)</td></tr>
<tr><td colspan="6">Other common courses</td></tr>
<tr><td colspan="4">(other mandatory options, 1 option)</td><td>MPS</td><td>CIT, SI …</td></tr>
<tr><td rowspan="4">Middle school (Collège)</td><td>9</td><td>Troisième</td><td colspan="7"></td></tr>
<tr><td>8</td><td>Quatrième</td><td colspan="7"></td></tr>
<tr><td>7</td><td>Cinquième</td><td colspan="7"></td></tr>
<tr><td>6</td><td>Sixième</td><td colspan="7"></td></tr>
</table>

Legend		
	ISN	Informatique & sciences du numérique (digital & computer science) mandatory option
	CIT / SI	Conception & Innovation technologiques / Sciences de l'ingénieur (technology-centric options)
	MPS	Méthodes et pratiques scientifiques (methods & practices in science)

How the system changes (2016-2019)

Levels	School years	French names	Particularities						
High school (Lycée)	12	Terminale	Math	Physics	Biology	ISN	Engineering	Techno	Vocational schools
			Common courses (math, physics-chemistry, life sciences, …)						
	11	Première	Science common courses (math, physics-chemistry, life sciences)					Techno	
			Other common courses (french, history & geography…)						
	10	Seconde	Science common courses (**math**, physics-chemistry, life sciences)						
			Other common courses						
			(other mandatory options, 1 option)			ICN	MPS	CIT, SI …	
Middle school (Collège)	9	Troisième							
	8	Quatrième							
	7	Cinquième							
	6	Sixième	**STEM lessons**						

Legend		
	ISN	Informatique & sciences du numérique (digital & computer science) mandatory option
	CIT / SI	Conception & Innovation technologiques / Sciences de l'ingénieur (technology-centric options)
	MPS	Méthodes et pratiques scientifiques (methods & practices in science)
		Algorithmics & Scratch programming (within math & technology courses)
	ICN	Informatique & création numérique (digital creativity & computer science) option

How the system changes (2019-2022)

Levels	School years	French names	Particularities						
High school (Lycée)	12	Terminale	Math	Physics	Biology	NSI	other options	Techno	Vocational schools
			Common courses (**science**, philosophy, history & geography…)					**math&physics**	
	11	Première	Math	Physics	Biology	NSI	other options	Techno	
			Common courses (**science**, french, history & geography…)					**math&physics**	
	10	Seconde	**Math (common course)**						
			Physics-chemistry (common course)						
			SNT (common course)						
Middle school (Collège)	9	Troisième							
	8	Quatrième							
	7	Cinquième							
	6	Sixième	**STEM lessons**						

Legend		
	NSI	Numérique et science informatique (digital science & informatics) mandatory option
	SNT	Sciences du numérique et technologie (digital science & technology course), **for all students**
		Algorithmics & Python programming
		Algorithmics & Scratch programming (within math & technology courses)

Trend 1 : teaching informatics

- About the curriculum
- Why Python?
- How to fail?
- Which technology?

Trend 1: Informatics (=computer science) (1)

- Computer science courses, from marginal and optional, are becoming widespread and (partially) mandatory in France.
- 2011→2016: Informatics is a (mandatory) option proposed to the pupils in grade 12 year (2h / week) (*).
- 2016→2019: Same + Informatics & digital creativity as a mandatory option proposed to the pupils in grade 10 year.
- 2019: Informatics (as SNT, “digital sciences”) becomes a mandatory course in grade 10, and mandatory option in grade 11 (4h / week).
- 2020: Same + informatics as mandatory option in grade 12 (6h / week).
- **Similar changes happen in some regions of Switzerland, England, Germany.**

Tren

- Computer
  widesprea
- 2011→201
  grade 12 y
- 2016→201
  proposed t
- 2019: Infor
  grade 10, a
- 2020: Sam
- **Similar cha**

Algorithmen

MI.2.2		Die Schülerinnen und Schüler ...	
1	a	» können formale Anleitungen erkennen und ihnen folgen (z.B. Koch- und Backrezepte, Spiel- und Bastelanleitungen, Tanzchoreographien).	
2			
	b	» können durch Probieren Lösungswege für einfache Problemstellungen suchen und auf Korrektheit prüfen (z.B. einen Weg suchen, eine Spielstrategie entwickeln). Sie können verschiedene Lösungswege vergleichen.	
	c	» können Abläufe mit Schleifen und Verzweigungen aus ihrer Umwelt erkennen, beschreiben und strukturiert darstellen (z.B. mittels Flussdiagrammen).	
	d	» können einfache Abläufe mit Schleifen, bedingten Anweisungen und Parametern lesen und manuell ausführen.	
	e	» verstehen, dass ein Computer nur vordefinierte Anweisungen ausführen kann und dass ein Programm eine Abfolge von solchen Anweisungen ist.	
	f	» können Programme mit Schleifen, bedingten Anweisungen und Parametern schreiben und testen.	MI - Produktion und Präsentation MA.2.C.2.g
3	g	» können selbstentdeckte Lösungswege für einfache Probleme in Form von lauffähigen und korrekten Computerprogrammen mit Schleifen, bedingten Anweisungen und Parametern formulieren.	
	h	» können selbstentwickelte Algorithmen in Form von lauffähigen und korrekten Computerprogrammen mit Variablen und Unterprogrammen formulieren.	
	i	» können verschiedene Algorithmen zur Lösung desselben Problems vergleichen und beurteilen (z.B. lineare und binäre Suche, Sortierverfahren).	

SNT (introduction to CS) course (grade 10)

Description of the SNT (digital sciences and technology) course

This course is an extension of the algorithms, computer science and programming parts of the mathematics and technology curriculum in the middle schools. The concepts and practices of programming are developed through activities related to the following themes:

Internet; the **Web**; **social networks**; **structured data** and their processing; **location**, mapping and mobility; **embedded** computing and **connected objects**; **digital imaging**.

Each of these themes is intended to be taught over a period of about four weeks.[…]

This course aims to create opportunities for various forms of activities (presentations, group work, mini-projects, individual or collective productions, etc.) developing transversal skills:

- demonstrate **autonomy, initiative and creativity**;
- present a problem or its solution, develop an argument in a debate;
- **cooperate** within a team;
- search and gather information, use **quality sources**, share resources;
- make responsible and **critical use** of digital sciences and technology.

Translated by Robert Cabane from the Official Bulletin of MoE

Source : https://cache.media.education.gouv.fr/file/CSP/41/0/2de_Sciences...

Trend 1: Informatics (=computer science) (2)

Consequences

- Grade 10 students will know how to code quite early
- Informatics now claims to be part of STEM
- The teachers will face many needs:
  - getting trained and prepared
  - dealing with large classes (up to 35 pupils)
  - lacking of devices and dedicated rooms
- Who will provide the best technology and the best training?

Why Python?

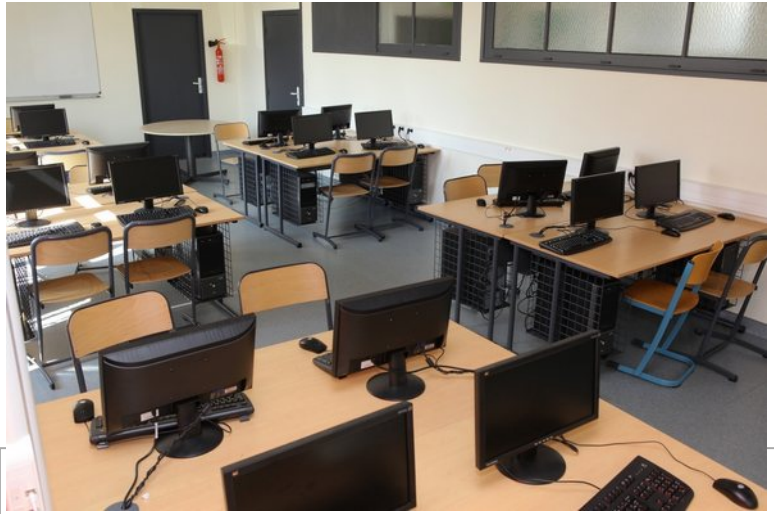
- A programming language was needed, the same for all schools when possible.
- Requirements :
  - interpreted (no hassles with the compiler)
  - concise (can be used to express algorithms even **on paper**)
  - widespread and well-defined (**truly open** standard)
  - universal (variety of environments and libraries)
- 3 candidates : Ruby, Python, Lua. Python won.

How **not** to teach teaching computer science

Now, the curricula are set, things are going to happen. Just tell to the school directors they have to organize things … and wait ?

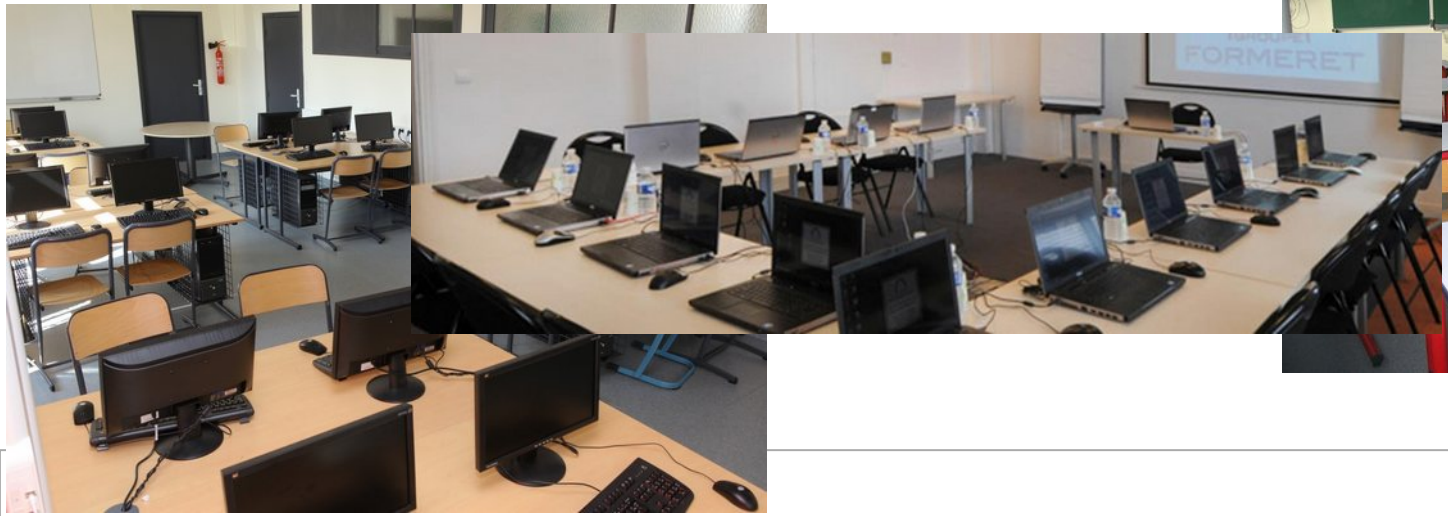
How **not** to teach teaching computer science

Now, the curricula are set, things are going to happen. Just tell to the school directors they have to organize things … and wait ?



How **not** to teach teaching computer science

Now, the curricula are set, things are going to happen. Just tell to the school directors they have to organize things … and wait ?



How **not** to teach teaching CS (7 ways to fail)

- You never have enough working (i.e. not broken) computers,
- and even if you had them, you wouldn't have enough rooms to install them,
- or the OS wouldn't boot in less than 5 minutes,
- and even if you had rooms and computers, you wouldn't have enough people to maintain them on a day-to-day basis,
- and even if you had laptops, you wouldn't have enough electrical sockets to connect them,
- and even so, the teachers would be mostly unprepared,
- and even the teachers acquainted with CS and digital technologies would be in need of pedagogy and didactics …

Technology for teaching computer science?

This is where calculator technologies would come in, with devices

- Highly autonomous
  (36 hours at least)
- Light but robust
- No distractions included
- Focused on science
- With controlled connectivity

Technology for teaching computer science?

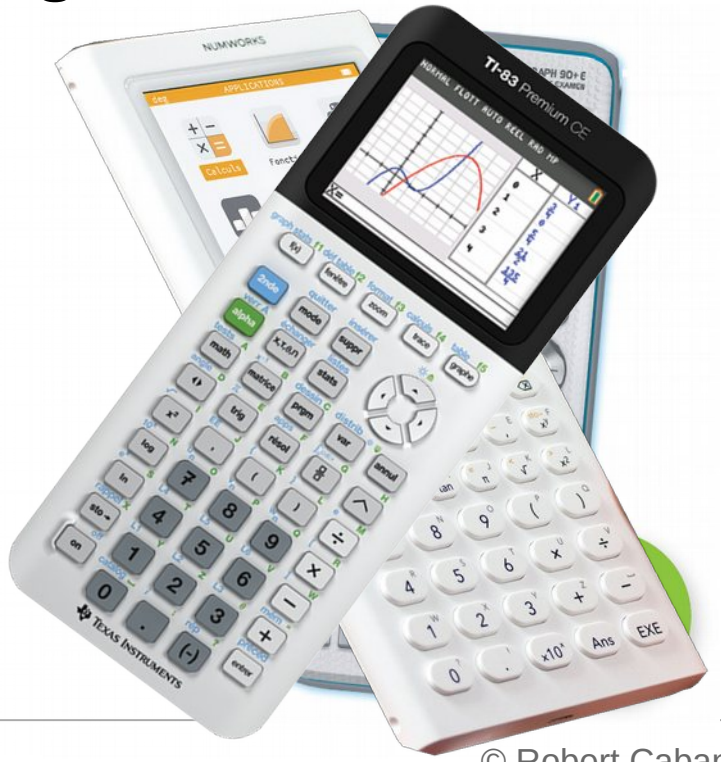
This is where calculator technologies would come in, with devices like this :

- Highly autonomous (36 hours at least)
- Light but robust
- No distractions included
- Focused on science
- With controlled connectivity

Technology for teaching computer science?

This is where calculator technologies would come in, with devices like this :

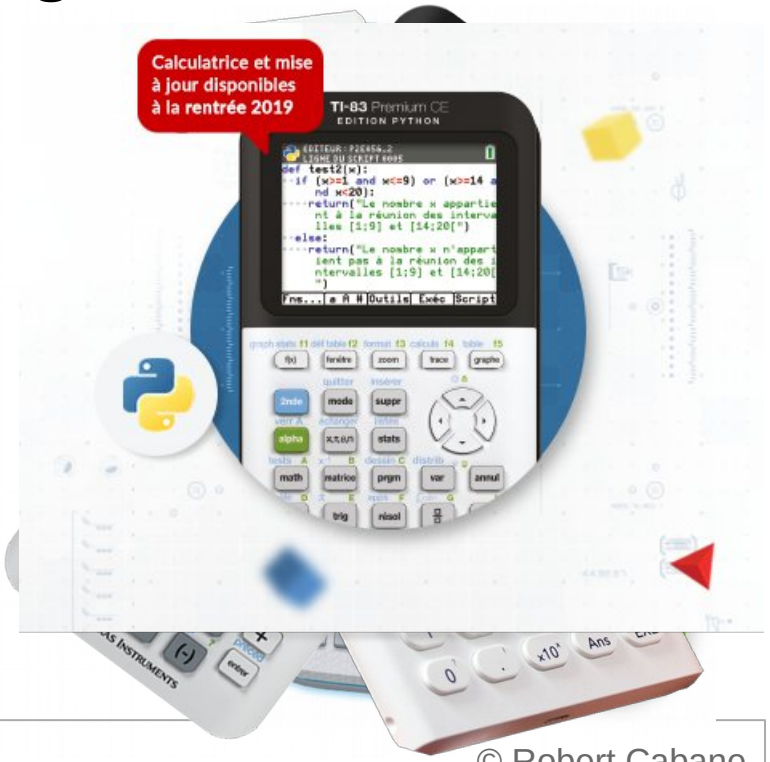
- Highly autonomous (36 hours at least)
- Light but robust
- No distractions included
- Focused on science
- With controlled connectivity



Technology for teaching computer science?

This is where calculator technologies would come in, with devices like this :

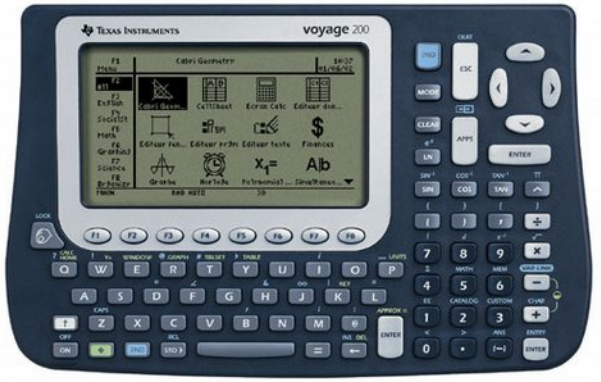
- Highly autonomous (36 hours at least)
- Light but robust
- No distractions included
- Focused on science
- With controlled connectivity



Technology for teaching computer science?

This is where calculator technologies would come in, with devices or like that ?

- Highly autonomous
  (36 hours at least)
- Light but robust
- No distractions included
- Focused on science
- With controlled connectivity
- **With a real keyboard and not-too-small screen**



Discussion

- Do you / would you include informatics among the STEM courses?
  - as part of math/physics/engineering courses
  - or as a specific, independent course ?
- How do you / would you get the required teachers?
  - training them (how) ?
  - or hiring them ?

Trend 2: STEM-inspired teaching

- About the curriculum
- Training the teachers
- Discussion

About the curriculum (1)

First STEM implementation style: two connected curricula

- First example : informatics in the middle school

Grades 7-9 math curriculum	Grades 7-9 technology curriculum
Pupils discover the programming by developing simple programs in a project approach. Creating programs, they develop programming methods, revisit the notions of variables and functions in a different form, and practice reasoning.	This course aims to explain the digital solutions that drive the evolution of technical objects of the pupil's living environment. Algorithms are studied jointly in mathematics and technology.
Objectives	
Write, test, execute, correct a simple program.	Write, test, execute, correct a program. Under stand how a computer network works.
Competences and skills	
Break down a problem into sub-problems to structure a program; recognize patterns. Write, test, correct and execute a program in response to a given problem. Write a program in which actions are triggered by external events. Concept of computer variable. Sequences of instructions, loops, conditional instructions.	Analyze the expected behavior of a real system, break down the problem into subsystem problems in order to structure a control program. Write, test, correct and execute a program controlling a real system, check the expected behavior. Write a program in which actions are triggered by external events.
Recommended programming environment: Scratch (or similar)	

About the curriculum (1)

First STEM implementation style: two connected curricula

- Second example : grade 10 math & informatics

Grade 10 math curriculum	Grade 10 SNT curriculum
Consolidation of middle school achievements around two essential ideas: • the notion of **function**; • **programming** (coding) as the production of a text in a computer language.	Extension of middle school achievements. Concepts and practices of programming are developed through activities related to: **Internet**; the **Web**; **social networks**; **structured data** and their processing; **location**, mapping and mobility; **embedded** computing and **connected objects**; **digital imaging**. Each these themes should be taken for about four weeks.
Tasks • describe algorithms in natural language or in a programming language; • realize some of these algorithms using a simple program written in a textual programming language; • interpret, extend or modify more complex algorithms.	This course aims to create opportunities for various forms of activities (presentations, group work, mini-projects, individual or collective productions, etc.) developing transversal skills: • demonstrate **autonomy, initiative and creativity**; • present a problem or its solution, develop an argument in a debate; • **cooperate** within a team; • search and gather information, use **quality sources**, share resources; • make responsible and **critical use** of digital sciences and technology.
Programming langage: Python	

About the curriculum (2)

Second STEM implementation style : one STEM curriculum, two (or more) teachers

- Example: grade 6 STEM course

The planet Earth and the action of the human mankind on its environment
Knowledges, targeted competences

Locate the Earth in the solar system solar and characterize the conditions of life on Earth (**temperature, presence of liquid water**).
Describe the **move of the Earth** (rotation about itself with the day-night cycle, around of the Sun, seasons cycle).
Use **geometric representations** of the space with some celestial bodies (using circles and spheres).
Identify the **organic and geological components** of a landscape.
Locate some opportunities available through the **geology** (groundwater, mines...).
Geological phenomena provoked by the internal activity of the Earth (volcanism, earthquakes, …).
Phenomena reflecting the external activity of the Earth (meteorological and climatic occurrences like storms, cyclons, floods and droughts).
Risks, necessity to protect the population.

Reference: http://eduscol.education.fr/...

About the curriculum (2)

Second STEM implementation style : one STEM curriculum, two (or more) teachers

- Second example: grade 11 sciences course

Ordered structures: crystals

The molecular organization being already known, this theme addresses another form of organization of matter, the crystalline state. Understanding this organization through the selected examples mobilizes knowledge about the **geometry of the cube.** It also provides an opportunity to develop skills in spatial representation and **volume calculations**.

Knowledge	Know-how
Solid sodium chloride (present in rocks, or resulting from the evaporation of seawater) consists of a regular **stack of ions**: it is the crystalline state. More generally, a crystalline structure is defined by an elementary mesh repeated periodically. A crystalline type is defined by the geometric shape of the mesh, the nature and position in this mesh of the entities that constitute it.	Connect the organization of the mesh at the microscopic level to the structure of the crystal at the macroscopic level. For each of the two networks (simple cubic and face-centered cubic): • represent the mesh in **perspective**; • calculate the compactness in the case of tangential spherical entities; • count the atoms per mesh and calculate the **density of the crystal**.

Reference: http://eduscol.education.fr/...

Training the teachers, new problems

- Being able to **use** mathematics / informatics isn't the same as being able to **teach** it.
- French math teachers usually have poor knowledge of physics (and even worse about chemistry, biology …).
- Math teachers don't care much about experimentation.
- Technology and physics teachers usually think that using math is easy for everyone.
- Different teachers, different **cultures**...

Training the teachers, inefficient methods

- excessive confidence on the textbooks
- big teachers meetings / conferences
- self-organization
- tutorials on the web (and nothing more)
- others ?

Training the teachers, better methods

- combine approaches (meetings+MOOCs+textbooks)
- don’t leave the teachers “alone in the dark”
- drive them “to the other side”
  - math teachers should discover experimental physics (not only conceptual, mathematical physics)
  - physics teachers should deal with abstraction
- work about language (terminology, notations)

Discussion (1)

- Would you prefer STEM-specific courses, or reshape existing science courses ?
  - Describe your experiences
  - Make proposals

Discussion (2)

- How would you train the teachers ?
  - Describe your favorite methods
  - Explain the outcomes
  - Make proposals