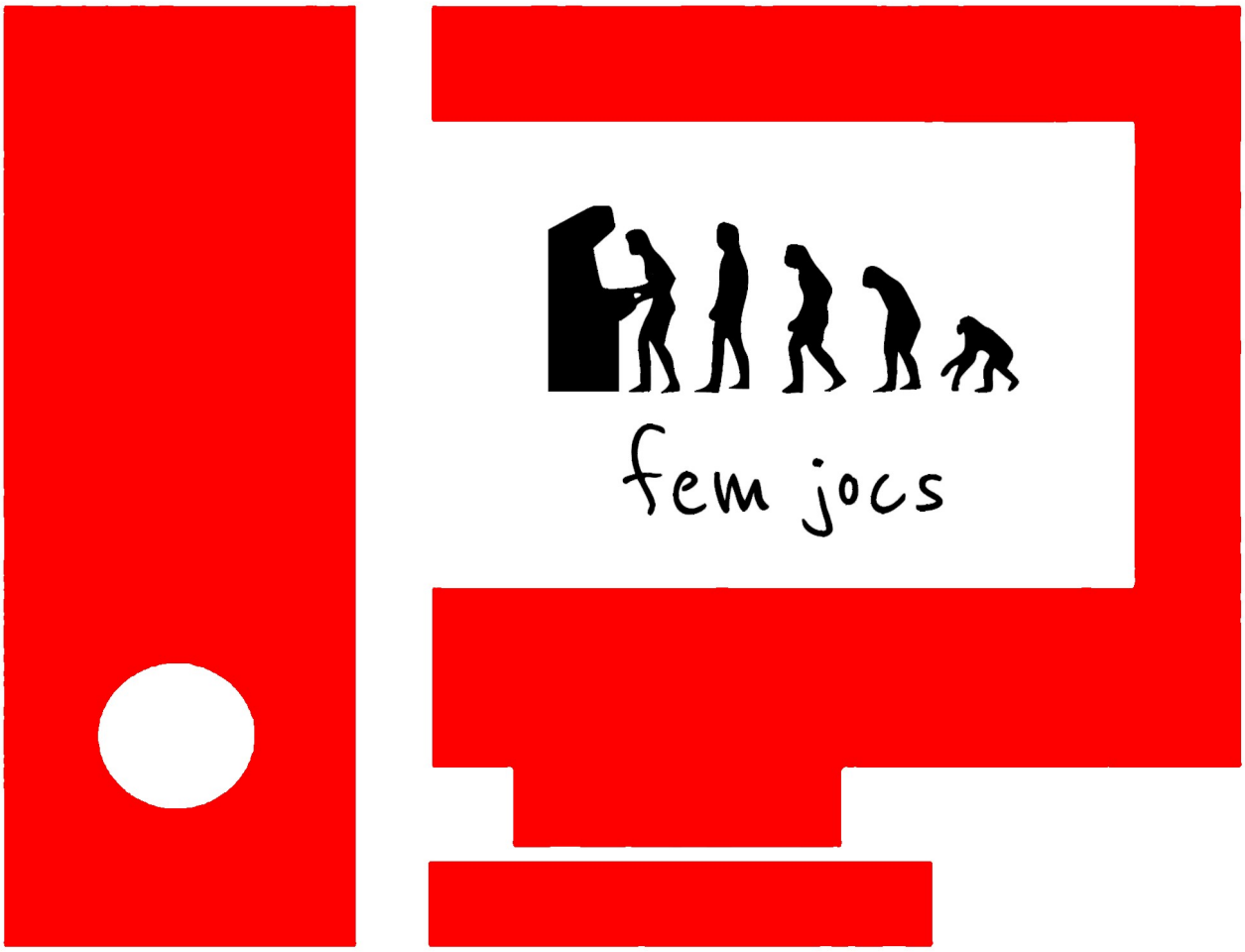
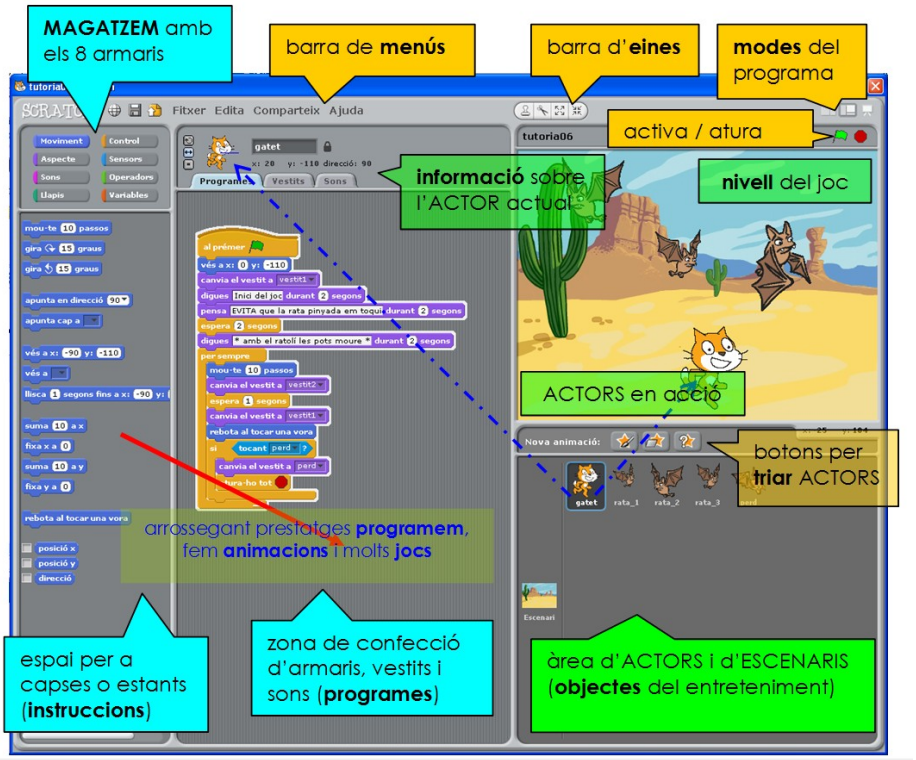
més coses a la web http://femjocs.com

À-SER REPARAPCS



Fem jocs, assolim estrelles
Endrecem bits, fabriquem mons i construïm herois amb SCRATCH

Aquesta aplicació ens permet fer animacions o jocs d'una manera fàcil i estructurada sent la forma **més senzilla** d'introduir-se en el món de la programació doncs, escolars de **8 anys**, poden fer-ho intuïtivament sense cap coneixe-ment previ. Aquest Scratch és un llenguatge de programació base que **ens facilita la creació d'històries interactives** perme-tent-nos, a més, el compartir-ho **via web** amb d'altres



Aquesta aproximació la farem amb la versió 1.4 però, l'entorn de treball en Xarxa ja disposa d'una superior, la versió 3, que permet entre altres coses l'edició de projectes en línia amb noves categories o armaris dins del magatzem, el canvi de nom de la que nosaltres anomenarem Variables i tot un paquet de nous prestatges per a la majora dels nostres armaris

Aquests jocs, des de l'Scratch, discorren per un o varis nivells que anomenarem **escenaris**. Els hem de crear, són plans i poden contenir tot tipus d'elements i objectes. Per confeccionar-los només hem de dibuixar-los o important-los com a fons doncs, un quadre buit, només és això: un espai en blanc per emplenar situat a la part superior dreta de l'aplicació que farà de nivell pel joc

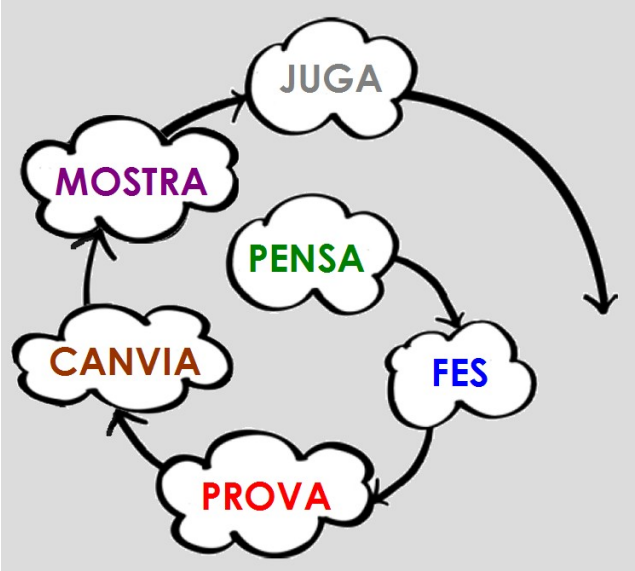
Per començar hem de veure que la majoria de coses que corren per una pantalla necessiten d'un suport gràfic, és a dir, d'una imatge. Aquí en direm **actors** i, per començar, apuntarem que es tracta d'un arxiu gràfic que pot contenir una o més imatges que són mostrades en forma de **vestit**

Un **actor**, en aquest cas, és un element al qual l'hi succeeixen coses anomenades Programes que arrossegarem des de la zona del **magatzem** en forma d'**estant acolorit, prestatge o lleixa** amb una balda que fa d'encaix per formar estructures noves: instruccions en forma d'**armari** que farà servir cada actor per desenvolupar-se

Aquestes pàgines volen ser una petita **aproximació** a la **programació** per a **mares**, **pares** i **tutors**. Doncs, des d'aquestes i en un llenguatge planer, explicarem els fonaments introductoris que s'han de conèixer perquè tothom pugui fer animacions, tutories i jocs... i quan diem tothom, és tothom, perquè només cal disposar d'una mica de

temps i empenta per, des del Cicle Inicial, dir també allò de **Fem jocs**!; per acabar depurant-los amb d'altres germans grans seus

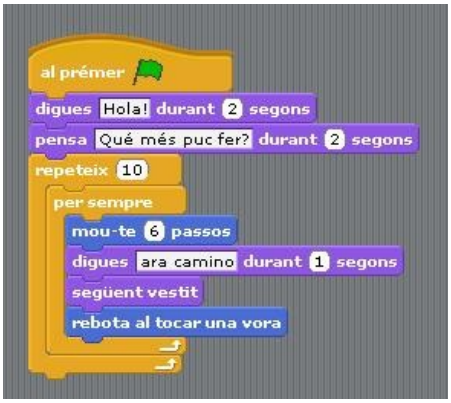
Conceptes bàsics de la programació



Som conscients que aquests conceptes a vegades carregosos i complexes fan allunyar-nos de l'ordinador per trobar que, tanta terminologia telemàtica, ens sobrepassa. Aquí demostrarem que això és ficció i que **qualsevol persona** pot esdevenir **programador** o **programadora** de videojocs –d'estar per casa, sí; però cal no oblidar que normalment és a casa on juguem i que per això no hem de tenir cap tipus de recança ni por perquè no competim, ni ens fixem límits o resultats doncs aquesta recerca en forma d'entreteniment ens proporcionarà coneixements, diversió i la possibilitat de compartir una activitat tecnològica prou pedagògica amb els nostres infants: feu la prova!

1. **La instrucció** = **prestatgeria acolorida**

En diem així de cada **ordre** que rep la màquina i la transforma en una **acció**. **Aquí les instruccions estan formades per acolorides lleixes o prestatges que emplenen els armaris del nostre magatzem**... dit així sembla una broma, però imagineu un espai com ara un magatzem amb de 8 armaris diferents i que cadascun guarda capses o estants que poden, depenent del cas, encaixar amb d'altes, no fer-ho o encabir-hi dades al interior per poder formar altres estructures noves que seran utilitzades pels nostres actors; accions que prendran la forma de programes i que els portaran a desenvolupar-se tal i com pretenem



2. **El programa** = **armari nou que utilitza l'actor o l'escenari**

Un programa no és ni més ni menys que el **resultat final** de totes les instruccions embolcallades dins d'un executable que durà a terme una finalitat concreta. Per exemple, el programa d'una Calculadora, tindrà totes aquelles rutines (conjunt d'instruccions per a cada element) que el permetran fer operacions aritmètiques com ara sumar, restar, multiplicar o dividir d'igual forma que passa amb els objectes i actors de l'Scratch. No cal dir que cada una d'aquestes operacions base s'hauran programat via instruccions per separat i se les hi haurà assignat, en aquesta mostra, un botó hàbil per tal de fer-ho tant funcional com entenedor per acabar rebent el nom de projecte o guió (botó de la calculadora = actor de l'Scratch –tenint present que sempre podrà desenvolupar més d'un armari o programa segons el moment què, si és inicial, ho farà al prémer la bandera verda; o

si és per moure's al prémer la tecla programada). D'aquesta manera **cada actor** nostre **utilitzarà aquests nous armaris, rutines o programes per assolir la finalitat** desitjada: emetre un so, rebotar en topar amb una vora o canviar-se de vestit

3. L'Estructura = forma del nostre armari

Cada llenguatge de programació disposa de la seva pròpia estructura, és a dir, manté una **forma concreta** de posar cada instrucció que ha de completar rutines per formar cada programa. Amb l'Scratch, la forma és simple, doncs sempre s'hauran d'iniciar, des de la pestanya Programes, amb **una lleixa arrodonida i groga del Armari de Control**; sobre la qual no hi podrà haver-hi res doncs l'encaix és dessota i, a partir d'aquí, només caldrà anar posant estant rere prestatge per formar el nostre particular armari i aconseguir el resultat que cerquem amb l'objecte que en farà ús

4. L'Estructura d'un algoritme = seqüència del nostre armari fet programa bàsic

D'igual forma que esmentàvem la calculadora com a exemple de programa global que incloïa d'altres de més petits aquí ens referirem als **diferents procediments que podem seguir per muntar el nostre armari** o, dit d'una altra manera, ens fixarem en les **possibles seqüències** de la mà d'un altre exemple quotidià: rentar-se les dents

Tots sabem com fer-ho... però, ens hem fixat en quin és el millor procés per dur-lo a terme?, quines són les seqüències més adients?, quin ordre és millor?, podem optimitzar-ho?: veiem-ho!

Rentar-se les dents, procediments comparatius		
Agafar el raspall	Agafar el raspall	Agafar un got
Obrir el tub de la pasta	Obrir l'aixeta de l'aigua	Obrir l'aixeta de l'aigua
Posar la pasta al raspall	Remullar el raspall	Omplir el got fins a dalt
Obrir l'aixeta de l'aigua	Tancar l'aixeta de l'aigua	Tancar l'aixeta de l'aigua
Remullar el raspall	Obrir la pasta de dents	Remullar el raspall
Tancar l'aixeta de l'aigua	Posar la pasta al raspall	Obrir la pasta de dents
Passar el raspall amb la pasta per les dents durant tres minuts	Passar el raspall amb la pasta per les dents durant dos minuts	Posar la pasta al raspall
		Tancar la pasta de dents
Obrir l'aixeta de l'aigua	Obrir l'aixeta de l'aigua	Posar el raspall a la boca
Esbandir-se la boca	Netejar el raspall	Rentar-se les dents
Netejar el raspall	Esbandir-se la bola	Esbandir-se la boca amb el got ple d'aigua tot glopejant
Tancar l'aixeta de l'aigua	Tancar la pasta de dents	Buidar el got d'aigua i deixar que s'eixugui dessobre la pica
Tancar la pasta de dents	Mirar-se al mirall per revisar el resultat de la neteja	

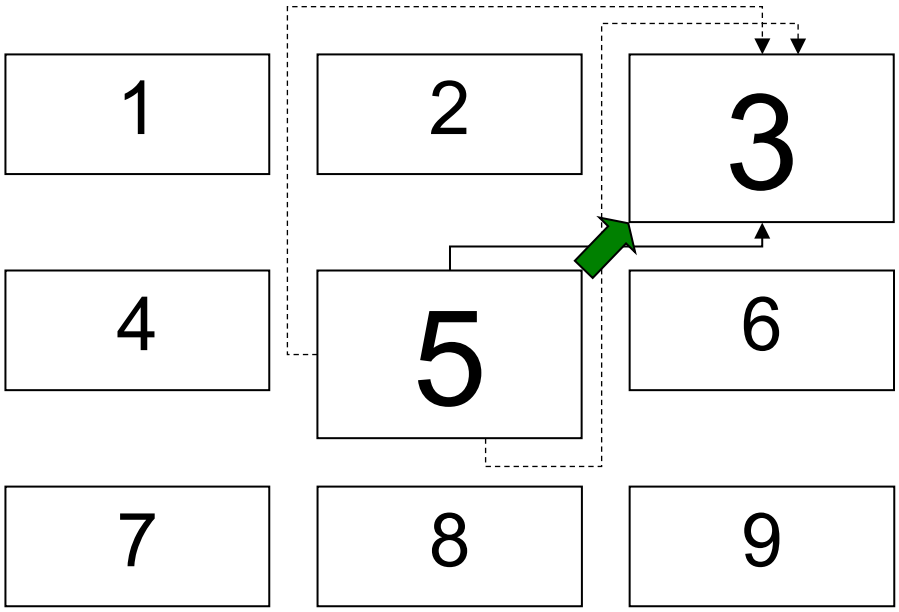
Desar tant el raspall com el tub de la pasta dentífrica

Després de veure l'anterior quadre quina de les tres opcions us sembla més adequada i quina menys (mireu de raonar la resposta oferint aportacions que us afegeixin cert valor i recordant que cada procés està format per 12 instàncies que corresponen a cadascuna de les cel·les; sent la que fa 13 comú als tres). Tanmateix, us sembla que hi hauria alguna possibilitat de millora prenent-ne una com a mostra i complementant-la amb aquelles instàncies que hi falten?, com ho definiríeu?, l'ordre "passar el raspall amb la pasta per les dents" podria ser una nova seqüència d'indicacions?, penseu-hi!

Acabem d'establir, potser sense saber-ho, allò què en matemàtiques i, per extensió, a informàtica s'anomena **algorisme** (sèrie ordenada d'instruccions, passos o activitats que ens porten a la solució d'un determinat problema, aquí, rentar-se les dents)

Anem a veure un altre exemple. Ens trobem al quadre central d'aquests nou que volen significar un plànol i ens pregunten com anar, pel carrer, al marcat com a número 3. Si veiem la imatge podrem trobar més d'una solució per anar-hi, aquí, més d'un camí que haurem de fer seguint unes indicacions o instruccions ordenades començant per fer la primera acció i després la immediatament següent fins a arribar a la destinació. Aquestes, les podrem representar a una llista, a una taula com en el exemple anterior del raspall (rentar-se les dents) o bé amb un dibuix anomenat diagrama i poden ser tant extenses com vulguem sempre que aconseguim el resultat o bé la solució que cerquem o ens indiquen. Haurem de veure però que el millor, per ràpid i òptim, sempre serà el més curt representat al document per una fletxa verda

Plànol de situació. Aquí cal resoldre l'anar del quadre número 5 al quadre número 3 en una seqüència ordenada

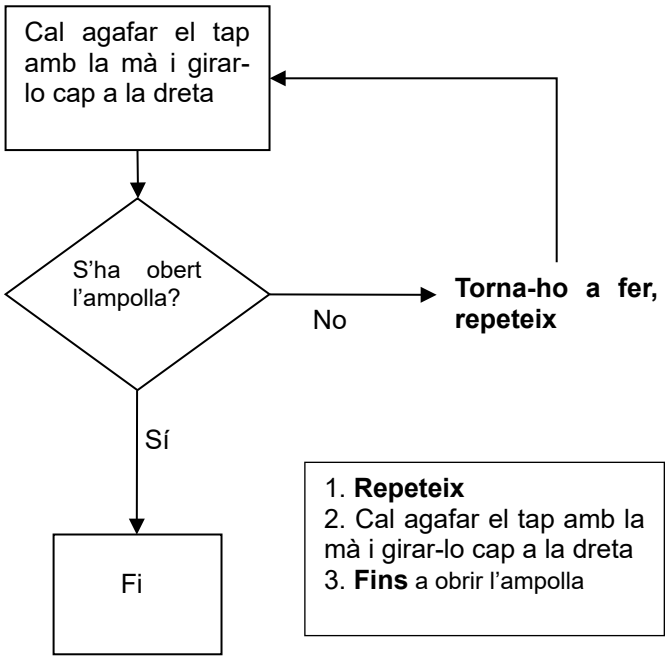


1. Sortir cap al nord
2. Girar cap a l'esquerra
3. Caminar fins a la següent illa de cases
4. Girar cap a la dreta
5. Caminar fins al destí

Com hem vist, aquestes indicacions també podrem representar-les a una llista –anomenada de pas a pas– que utilitzarem per dur a terme la tasca tot seguint la seva numeració. És molt important veure que donem moltes coses per suposades en els nostres cinc passos –i que per tant no expliquem– com ara per on sortim del quadre 5, els conceptes de direcció o sentit, que sabem què és caminar, següent o una illa de cases i girar més destí. Així doncs, a l'hora d'enfrontar-nos al disseny d'un programa en forma de

llista numerada cal tenir previstes i estudiades totes aquestes circumstàncies prèviament i per separat, tot provant de solucionar cada pas abans de provar fer el següent (cal observar també, per exemple, que diem "girar cap a" però no referim si haurem de fer mitja volta o bé una de sencera tot donant per suposat que aquí "girar cap a" vol dir fer-ho, de quart en quart, tenint en compte el sentit marcat: ull viu amb això!)

Un tercer. El problema que aquí plantegem serà destapar una ampolla d'aigua amb tap de rosca. Si hi penseu, ràpidament veureu que una de les instruccions, ordres o indicacions hauran de ser la mateixa repetida, no? Mirem aquest diagrama i les llistes que representen l'acció, per a nosaltres seqüència de l'armari dins l'Scratch, d'obrir un tap a rosca d'una ampolla tancada

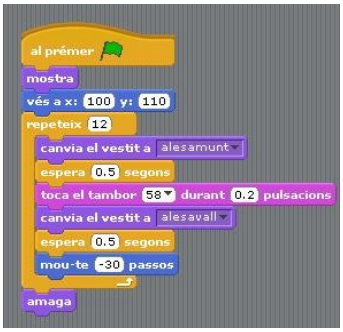


1. Cal agafar el tap amb la mà i girar-lo cap a la dreta
2. Deixa anar el tap
3. Cal agafar el tap amb la mà i girar-lo cap a la dreta
4. Deixa anar el tap
5. Cal agafar el tap amb la mà i girar-lo cap a la dreta
6. Deixa anar el tap
7. Cal agafar el tap amb la mà i girar-lo cap a la dreta
8. Deixa anar el tap

1. **Repeteix**
2. Cal agafar el tap amb la mà i girar-lo cap a la dreta
3. **Fins** a obrir l'ampolla

1. Cal agafar el tap amb la mà i girar-lo cap a la dreta
2. *No* s'ha obert l'ampolla?
3. **Repeteix, torna a l'1**

Com veiem és primordial que les instruccions siguin tant precises i senzilles com puguem per tal que qualsevol persona (per a nosaltres, dins de l'Scratch, ACTOR) les desenvolupi sense equívocs doncs, d'aquesta manera, optimitzarem tota la tasca sense que el resultat es vegi modificat. Proveu, en aquest moment, d'explicar la seqüència a seguir per fer un avió de paper, vestir-se un cop ens hem dutxat o per imprimir un document des d'un ordinador que està apagat atenent a la facilitat de les paraules que utilitzem i a la seva significació, doncs haurem de ser concrets i molt precisos si volem que qualsevol persona reprodueixi aquests tres fets sense errors (les estructures poden ser tant llargues com vulgueu doncs extens no és sinònim d'incorrecte però, un cop realitzades, hauríeu de millorar-les perseguint l'optimització del procediment: finalitat última de qualsevol programació). Així doncs l'estructura i seqüència de les nostres instruccions ens portarà a desenvolupar cada programa atenent al resultat que cerquem i, aquestes, es reproduiran l'una darrera de l'altra respectant tant l'ordre de construcció com la funcionalitat pròpia de cada bloc. Només es tracta de trobar les possibles solucions als reptes que, de mica en mica, ens anirem plantejant (el següent quadre mostra una de les possibles solucions, en forma d'algoritme, al problema que us plantejàvem)

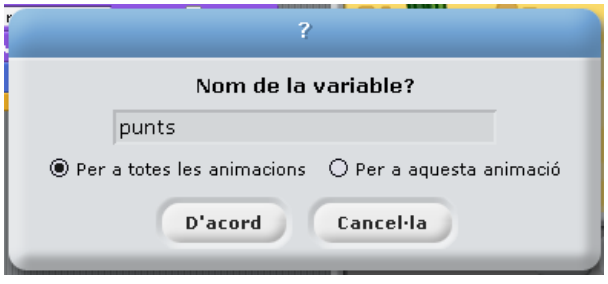


	Rentar-se les dents, possible algoritme
1	Agafar la pasta de dents i destapar-la
2	Agafar el raspall de dents
3	Posar la pasta de dents al raspall i tapar el tub de la pasta
4	Obrir l'aixeta del rentamans
5	Remullar el raspall amb l'aigua
6	Tancar l'aixeta del rentamans
7	Rentar-se les dents amb el raspall durant 3 minuts
8	Obrir l'aixeta del rentamans
9	Esbandir-se la boca
10	Netejar el raspall
11	Tancar l'aixeta del rentamans
	Assecar-se la cara i les mans amb una tovallola

5. La **variable**, el **bucle** o **tirabuixó**, la **constant**, els **operadors** i les **llistes**

Per completar aquesta petita introducció, força elemental, al món de la programació cal parlar d'aquests aspectes fonamentals:

Variable = **incògnita en forma de valor**



Aquí la definirem dient que **es tracta d'una capsa o contenidor al qual l'enganxem una etiqueta identificativa davant, per saber el que hi guarda, i que servirà per desar una cosa al interior** –segur que heu sentit dir alguna vegada que aquesta tindrà el valor X, Y o Z; doncs això, un valor per assignar. De variables, en el món de la programació, n'hi ha un munt i es diferencien pel seu tipus que definim abans de poder-les utilitzar… nosaltres, dins de l'Scratch, les crearem d'igual forma però ho farem des del botó que mostra l'Armari amb les peces taronges anomenat Variables
Aspectes dels jocs com ara **Vida**, **Salut**, **Energia** o **Puntuació** en serien; però no només aquests sinó que en sentit ampli dins d'aquestes caixes podem encabir-hi d'altres coses (fixes o no) com ara dades **lògiques**, **numèriques**, de **caràcter** o **tipus cadena** sigui per valor o bé per freqüència dins l'àmbit local –per a aquesta animació– o més global –per a totes les animacions– (aquesta *parrafada*, que segurament no s'entén, vol dir que podem posar-hi, per exemple, un SI o un No i anar-ho a mirar quan programem. Imaginem una situació on ens calgui una clau per obrir una porta i que l'haguem definida com a variable amb l'etiqueta tinc_clau; quan el nostre actor topi amb un objecte gràfic amb aquesta forma farem que la variable tinc_clau sigui SI permetent-nos accedir-hi, per en cas contrari, mantenir amb un NO la porta tancada prèvia comprovació del seu valor per continuar –físicament aniríem a la capsa i miraríem a dins per esbrinar el valor d'aquesta incògnita)

Bucle i **tirabuixó** = **repetició de cicles**

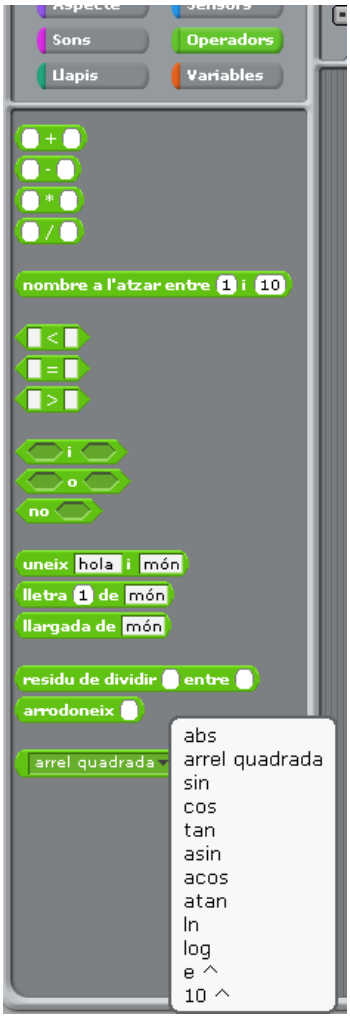
Podem definir-los genèricament com a **seqüència d'ordres aïllades que poden ser executades repetidament i direm que són instruccions especials en forma de serjant que n'abracen d'altres** –sent un dels fonaments de la programació estructurada més moderna. Aquesta mena de mordassa, que trobarem al Armari de Control de l'aplicació, actua segons els casos com una sola lleixa reproduint tot allò que la conforma (grup de prestatges) bé sigui per sempre, un nombre determinat de vegades o de manera condicional en estructures com ara: si passa això, fes el següent o; si no, el que ve a continuació

Com no podria ser d'una altra manera també es poden concatenar per formar altres estructures més grans i aquesta formació seria el que anomenaríem tirabuixó, fent referència a la circulació constant que fan els valors quan s'enfronten amb ell (una altra nomenclatura les anomena iteracions o loopings). Totes aquestes construccions estan basades en condicions que es duran a terme atenent al següent:

a) el bucle **es repeteix mentre** es compleix una condició
b) el bucle **es realitza fins** que s'acompleixi allò altre que diem

Constant = **dada invariable**

Aquí parlaríem d'una quantitat o peça alfanumèrica que **manté sempre** el mateix valor i que mai canvia durant l'execució del programa –es pot considerar com **una mena de contenidor fix**. En moltes ocasions es fan servir per alliberar l'ús tant de la memòria com del processador de la màquina, doncs limitats, convé utilitzar-los correctament i amb la major eficiència possible. Aquestes dades representen típicament valors matemàtics com ara el nombre Pi, que un cop definit o igualat a 3,1416; serà canviat en temps de compilació per aquest valor, és a dir, que quan l'haguem posat dins d'una capsa i etiquetat com a constant Pi –cada vegada que l'utilitzem a una fórmula– l'ordinador no l'haurà d'anar a cercar només el canviarà sent això un procés molt més àgil que l'ús propi de les variables



Operador = **càlcul matemàtic**

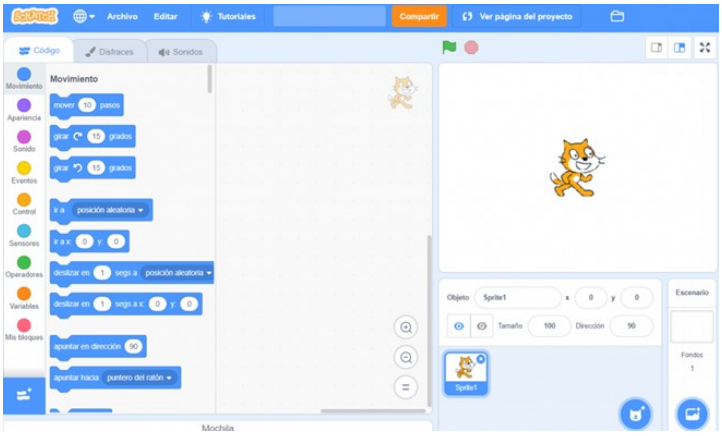
Els operadors serien aquelles peces que ens permetrien fer càlculs **aritmètics**, **lògics** o **relacionals** amb dades

Arrays = **llistes acumulatives**

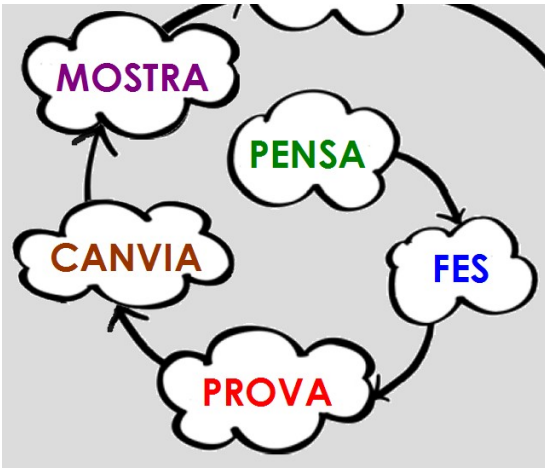
Existeixen una sèrie específica d'instruccions que ens permeten fer grups amb entrades tipus taula tot emmagatzemant els conceptes alfanumèrics com si féssim una llista, és a dir, una mena de col·lecció dinàmica amb dades a les que assignem una posició concreta per poder-les utilitzar. Fent un paral·lelisme, una mica bèstia, vindria a ser com la preparació de la llista de la compra, on anem posant element rere element en un paperet sense un ordre aparent on cada cosa a comprar és única i ocupa una posició en el llistat; podent d'una ullada veure'n un de concret o referint-nos-hi doncs, res ens impedeix, poder comprar el situat a la tercera posició primer (l'acte de comprar aquí seria sinònim d'acció programàtica)

Amb aquests últims conceptes tanquem l'acostament a la programació que us proposem pensant que ensenyar tot jugant amb eines informàtiques no és tant complicat sent, el propi Scratch, una bona mostra avalada pel prestigiós MIT i pels principis què, copiats de Xerox, aplicaren també els fundadors d'Apple a la seva màquina/sistema cap al 1982: **imagina > crea > experimenta**

Som molt conscients que amb això no n'hi ha prou i que, potser *deliberadament*, oblidem altres aspectes importants com ara la gestió dels moments, les entrades via teclat, la coordinació, l'execució paral·lela, la sincronització o la interacció dinàmica amb el ratolí; aspectes que tractarem, per separat, a moltes de les nostres propostes què en forma de repte, exemple o tutoria també us oferirem al nostre lloc femjocs.com



Per altra banda també cal apuntar que aquesta aplicació –des del punt de vista propi de la programació i tenint en compte que ja es troba a la seva versió 3– justeja en quan a conceptes que encara no contempla però què, de ben segur, abordarà a properes edicions com els procediments i les funcions, la recursivitat, el tractament d'excepcions, els paràmetres i els valors de retorn o la definició pròpia de classes



Passos a seguir per poder fer un videojoc

En aquesta introducció ja hem vist aquest quadre que us adjuntem a la dreta i que hem titulat "programació base"; ara ens en servirem per poder explicar el procés creatiu que ens ocupa tot desglossant-lo en senzills apartats

1. **PENSA**

La primera cosa que necessitarem és **una idea** que estarà formada, com a mínim, pels següents punts:

- **tipus de joc**: us recomanem que per començar proveu de fer algun clònic d'un joc que us agradi, és a dir, una còpia més o menys original d'un altre que conegueu
- **nombre de jugadors**: proveu amb 1 i, si us en sortiu, afegiu-ne d'altres
- objectiu del mateix: pot ser quasi qualsevol cosa des de deslliurar-se d'uns enemics fins a rescatar a una princesa, passant per tot aquell imaginari que pugueu ara recordar en tipologies que abracen tot tipus d'àmbits com les plataformes, els laberints, l'arcade o el rol
- **història o guió**: ha d'implicar als jugadors propiciant l'emoció i l'aprenentatge per nivell
- **disseny dels actors**: els objectes i els recursos han de ser els necessaris i els més planers possibles ajudant al desenvolupament del propi joc
- **disseny dels escenaris**: aquests seran els responsables d'oferir-nos tant el grau de dificultat com la presentació o cloenda del propi entreteniment
- **sons en interacció**: en diem així de tot so assignat als actors de manera directa
- **música de fons**: serà aquella que ens acompanyarà durant el transcurs del joc
- **programació**: què és el que necessito fer, a cada pas, per obtenir el resultat que cerco?

2. **FES**

Aquí és on el pes de tot plegat ens pot fer tirar enrere perquè toca treballar deixant enrere la fase de planificació per endinsar-nos cap a l'acció pròpia del entreteniment. En el nostre cas l'editor que ens ofereix l'Scratch serà indispensable per dibuixar-ho tot plegat i la seva llibreria gràfica un fons sense cost per utilitzar: feu-ne ús

3. **PROVA**

Després de programar-ho tot toca veure com va doncs és imprescindible poder testejar i provar el joc a diferents màquines i per quantes més persones millor. Cal veure en aquest moment que nosaltres, quan programem, llancem el joc tot provant-lo una

vegada i una altra convertint-nos en usuaris avançats i perdent la perspectiva… hauríem de fer-nos amb un cercle de provadors, amics incondicionals, que juguin amb l'entorn i ens ofereixin una visió molt més acurada: indicacions i consells que seguirem pel bon funcionament i desenvolupament del joc

4. **CANVIA**

Arriba l'hora de les millores fruit de l'estudi i de les proves efectuades, és a dir, de les revisions; doncs tota aplicació porta implícita la seva versió que no és més que el grau de maduresa de la mateixa. Quan es programa un joc o qualsevol altre software partim d'una versió beta o en fase de prova que esdevindrà candidata a la 0 i, a partir d'aquí, es succeiran les diferents propostes fins a deixar l'aplicació com a definitiva sense errors aparents i amb totes les millores incorporades

5. **MOSTRA**

La publicació pròpia del joc és la base per poder donar-se a conèixer i aprendre moltes més coses alhora que compartim la feina realitzada donant vida i possibilitats creatives a la resta d'usuaris de la Xarxa, programadores i programadors, que faran com nosaltres: el prendran com a model, realitzaran un seguit de canvis o millores i el tornaran a compartir