

Jasně! Tohle vložíš úplně na začátek toho předchozího worksheetu. Je to zaměřené na to, aby pochopili, že `local` není jen magické slovo, ale způsob, jak si něco "uložit na potom".

Níže máš **přídavný list pro děti** a pod ním **tvůj Answer Key (tahák)** ke všem otázkám (i těm z minula).



Část 0: Úplné základy (Proměnné a Funkce)

(Tohle vyplňujeme společně, než začneme programovat)

A) Proměnná (Variable) = Chytrá krabice 📦

Představ si proměnnou jako **krabici**, na kterou si nalepíš štítek se jménem a dovnitř něco vložíš. Počítač si to pak pamatuje.

V Lua se krabice vytváří slovem `local`.

Příklad kódu:

```
local mojeZlato = 50
local mojeJmeno = "Karel"
```

Úkoly:

- Jak se jmenuje krabice (proměnná), ve které je číslo 50?
 - Odpověď:
- Co je uvnitř krabice s názvem `mojeJmeno`?
 - Odpověď:
- Doplň kód:** Chci vytvořit proměnnou `pozdrav`, ve které bude text "Ahoj".
 - `local = "Ahoj"`

B) Funkce (Function) = Recept 📖

Funkce je seznam instrukcí (recept), který se provede až ve chvíli, kdy ho "zavoláme".

Příklad kódu:

```
local function udelejPalacinky()
    print("Rozklepni vejce")
    print("Pridej mouku")
    print("Osmaž to")
end
```

Otázky:

1. Když napíšu tento kód, udělají se palačinky hned? (ANO / NE)
 - Odpověď:
2. Co musím udělat, aby se kód uvnitř spustil? (Vyber jednu)
 - A) Smazat ho.
 - B) Zavolat tu funkci jménem (nebo ji na něco napojit).
 - C) Vypnout počítač.

Tady je worksheet navržený speciálně pro **začátečníky a pomalejší děti**. Je strukturovaný tak, aby si nejdřív společně zopakovali základy a pak šli krok za krokem k cíli. Text je jednoduchý, v bodech, aby se v tom neztratili.

Worksheet: Programujeme Smrtící Lávu (Kill Block)

Jméno programátora: _____

Část 1: Rychlé "Opačko" (Děláme společně na tabuli)

Doplňte chybějící slova do kódu nebo odpovězte na otázky.

1. Hierarchie (Rodokmen hry):
Když chci v kódu najít nějaký Part, který leží ve světě, začínám slovem game. Pak jdu do kufříku Workspace.
 - Celá cesta vypadá takto: `game._____MujSuperPart`
2. Vlastnosti (Properties):
Když chci změnit průhlednost (Transparency) na polovinu, napíšu:
 - `script.Parent.Transparency = _____` (jaké číslo je polovina?)
3. Událost (Event):
Aby se něco stalo, když se něčeho dotknu, používáme událost:
 - `Touched` (Dotek)
 - `Clicked` (Kliknutí)(Zakroužkuj správnou možnost)

Část 2: Vytváříme Lávu (Grafika)

Ted' si vyrobíme kostku, která bude vypadat nebezpečně.

1. **Vlož Part:**
 - V horním menu klikni na **Home** a vyber **Part** (kostka).
2. **Změň barvu a materiál:**
 - Klikni na Part ve hře.
 - V okně **Properties** (Vlastnosti) najdi `BrickColor` a změň ho na **Really Red**.
 - Najdi `Material` a změň ho na **Neon** (aby svítil).
3. **Důležité! Ukotvení:**

- Najdi v Properties políčko **Anchored** a **zaškrtni ho**.
 - *Proč?* Aby láva nespadla někam do díry, ale zůstala tam, kde má být.
4. **Přejmenování:**
- V okně **Explorer** najdi svůj Part a přejmenuj ho na **Lava**.

[ZDE VLOŽ SCREENSHOT: Šipka ukazující na tlačítko Part, pak ukázka okna Properties se zvýrazněným Anchored a Neon]

Část 3: Programujeme Smrt (Script) 💀

Ted' naučíme lávu, aby ubližovala.

1. Najdi v Exploreru svou **Lavu**.
2. Klikni na **plus (+)** vedle ní a vyber **Script**.
3. Smaž `print("Hello world!")`.
4. **Opiš tento kód přesně takto:**

Část 4: Vysvětlení pro zvědavé (Jak to funguje?)

Představ si to jako **zvonek u dveří**:

1. **lava.Touched**: To je tlačítko zvonku. Čeká, až ho někdo zmáčkne (dotkne se).
2. **Connect(zabitHrace)**: Tím říkáme: "Když někdo zazvoní, udělej to, co je v receptu `zabitHrace`."
3. **toCoSeDotklo**: To je informace, **co přesně** zazvonilo. V Robloxu se lávy dotkne nejdřív tvoje **noha** (LeftFoot nebo RightFoot).
4. **toCoSeDotklo.Parent**: My ale nechceme zabít jen nohu! Chceme zabít celého hráče. Proto se podíváme na **rodiče** té nohy -> to je celá **Postava (Character)**.
5. **humanoid.Health = 0**: Najdeme v postavě "Humanoid" (to je věc, která hlídá zdraví) a vypneme ho.

Část 5: Testování

1. Klikni nahoře na **Play**.
 2. Dojdi k lávě.
 3. Stoupni na ni.
 4. **Rozpadl jsi se?**
 - [] ANO (Jsi super programátor!)
 - [] NE (Zavolej lektorku, najdeme chybu v Outputu.)
-

Bonus pro rychlíky 🚀

Hotovo? Zkus tohle:

Změň skript tak, aby tě láva nezabila hned, ale jen tě zranila.

- *Nápověda:* Místo `humanoid.Health = 0` zkus napsat `humanoid.Health = humanoid.Health - 10`.
- Co se stane, když na lávě budeš stát dlouho?

Co dál pro tebe (Lektorku)?

Tento návod je hodně "blbuvzdorný", ale děti se často zaseknou na překlepech.

Nejčastější chyby, které budeš řešit:

- `script.parent` s malým P (musí být `Parent`).
- `humanoid.health` s malým H (musí být `Health`).
- Zapomenuté `end` na konci funkce.

1. Nejčastější bugy (Aneb "Proč mi to nefunguje?")

- **"Paní učitelko, láva zmizela!"**
 - **Příčina:** Zapomněli zaškrtnout **Anchored**.
 - **Co se stalo:** Jakmile spustili hru, fyzika zafungovala a neukotvená kostka propadla podlahou (Baseplate) do prázdnota.
 - **Oprava:** Stopnout hru -> Kliknout na Lávu -> Properties -> Zaškrtnout **Anchored**.
- **"Stojím na tom a neumírám!"**
 - **Příčina A (Umístění):** Skript není uvnitř Lávy (je třeba ve Workspace nebo v `ServerScriptService`).
 - *Fix:* `script.Parent` hledá rodiče skriptu. Pokud skript není v Lávě, `script.Parent` je něco jiného a `Touched` se nikdy nespustí.
 - **Příčina B (Velká písmena):** Napsali `humanoid.health` s malým `h`.
 - *Fix:* V Robloxu jsou vlastnosti vždy Velkým Písmenem. Musí to být `.Health`.
 - **Příčina C (Událost):** Napsali `Touch` nebo `touch` místo `Touched`.
- **"Mám v Outputu červený error: attempt to index nil with 'Health'"**
 - **Příčina:** Skript našel `humanoid`, ale ten je prázdný (`nil`), nebo se snaží pracovat s něčím, co neexistuje.
 - **Častější příčina u dětí:** Napsali `local humanoid = postava:FindFirstChild("humanoid")` (s malým `h`).
 - **Vysvětlení:** Jméno objektu v postavičce je "Humanoid" (velké H). `FindFirstChild` je citlivé na velikost písmen. Když ho nenajde, vrátí `nil` (nic). A "nic" nemá životy.

2. Technické "Proč" (Pod kapotou)

- **Proč ověřujeme `if humanoid then`?**
 - Událost `Touched` se spustí při doteku s *čímkoliv*.
 - Když do lávy spadne třeba míč nebo jiná kostka, skript se spustí.
 - Míč ale nemá `Parent` (postavu) a nemá `Humanoid`.
 - Kdybychom tam tu podmínku `if` neměli, skript by spadl s chybou pokaždé, když se lávy dotkne něco neživého.
- **Kulometný efekt (Debounce)**
 - Uvědom si, že `Touched` se nespustí jednou. Spustí se třeba **20x za vteřinu**, dokud se hráč hýbe a dotýká se partu.
 - U zabíjení (`Health = 0`) to nevadí (mrtvý je mrtvý).
 - U ubírání života (`Health = Health - 10`) by to hráče vypařilo okamžitě, protože by se to odečetlo 20x v mžiku. *To je důvod, proč se u damage bloků učí pokročilejší koncept tzv. "Debounce" (časový zámek), ale to děti zatím nezatěžuj.*

3. Otázky na tělo (Kritické myšlení)

- **Otázka:** "Zabije tahle láva i Zombie (NPC), které do hry vložíme z Toolboxu?"
 - *Odpověď:* **Ano.**
 - *Proč:* Protože Zombie v Robloxu jsou postavené stejně jako hráči – mají `Model` a uvnitř `Humanoid`. Náš skript nerozlišuje mezi hráčem a příšerou, hledá jen "něco, co má životy".
- **Otázka:** "Co by se stalo, kdybychom ten skript dali do StarterPlayerScripts místo do Partu?"
 - *Odpověď:* **Nefungoval by (nebo špatně).**
 - *Proč:* Protože `script.Parent` by nebyla Láva, ale složka PlayerScripts. Museli bychom v kódu lávu hledat složitě přes `game.Workspace.Lava...`
- **Otázka:** "Jak udělat, aby láva zabíjela jenom Tým 1 a Tým 2 nechala přežít?" (Pro ty nejrychlejší)
 - *Nápověda:* Museli bychom z `game.Players` zjistit, k jakému hráči ta postava patří, a zkontrolovat jeho `TeamColor`. (To je už ale vyšší dívčí).



ANSWER KEY

Zde máš správné odpovědi a stručné vysvětlení ("Proč"), kdyby se někdo ptal. Můžeš to číst přímo z papíru.

Odpovědi k Části 0 (Úplné základy - viz výše)

A) Proměnná

1. **Odpověď:** `mojeZlato`
 - *Vysvětlení:* To, co je za slovem `local`, je název krabice.

2. **Odpověď:** "Karel" (nebo prostě Karel)
 - *Vysvětlení:* To, co je za rovnítkem `=`, je obsah krabice. U textu musí být uvozovky!
3. **Odpověď:** pozdrav
 - *Vysvětlení:* Celý řádek je: `local pozdrav = "Ahoj"`.

B) Funkce

1. **Odpověď:** NE
 - *Vysvětlení:* Funkce je jen *návod* uložený v šuplíku. Dokud ji někdo nezavolá, nic se nestane.
2. **Odpověď:** B) Zavolat tu funkci...
 - *Vysvětlení:* Stejně jako kuchařka sama neuvaří večeři, dokud podle ní nezačneš vařit. V našem případě "vaření" spustí dotek lávy (Event).

Odpovědi k Části 1 (Rychlé "Opačko" z předchozího worksheetu)

Tohle jsou odpovědi k té části "Hierarchie" a "Vlastnosti", co jsem posílal v minulé zprávě.

1. Hierarchie (Rodokmen hry)

- Doplnovačka: `game.Workspace.MujSuperPart`
- *Vysvětlení:* Všechny fyzické věci (party, stromy, spawny) jsou vždy ve složce `Workspace`.

2. Vlastnosti (Properties)

- Doplnovačka: `0.5`
- *Vysvětlení:* V Robloxu je průhlednost (Transparency) číslo od 0 do 1.
 - 0 = Úplně vidět (plné).
 - 1 = Neviditelné.
 - 0.5 = Poloprůhledné (duch).

3. Událost (Event)

- Správná odpověď: **Touched**
- *Vysvětlení:*
 - `Touched` = fyzický kontakt (vrazím do toho).
 - `Clicked` (resp. `MouseButtonClick`) = kliknutí myší (používá se u `ClickDetector`, ne u lávy).

Rychlý tip, jak to vysvětlit dětem ("Koncept sendviče")

U toho skriptu s lávou jim to nejlépe vysvětlíš jako výrobu sendviče (funkce):

1. **Ingredience (Proměnné):** Nejdřív si připravím chleba a šunku (`local lava`, `local function`).
2. **Postup (Tělo funkce):** Dám šunku na chleba (`humanoid.Health = 0`).
3. **Hlad (Event):** Až když dostanu hlad (`Touched`), tak ten postup provedu.